

**SISTEM REKOMENDASI BUKU PERPUSTAKAAN  
FAKULTAS SAINS DAN TEKNOLOGI DENGAN METODE  
*COLLABORATIVE FILTERING***

**TUGAS AKHIR**

Diajukan Sebagai Salah Satu Syarat  
Untuk Memperoleh Gelar Sarjana Teknik Pada  
Jurusan Teknik Informatika

Oleh :

**ARITHA HANDRICO**  
**10851001718**



**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU  
PEKANBARU**

**2012**

**SISTEM REKOMENDASI BUKU PERPUSTAKAAN  
FAKULTAS SAINS DAN TEKNOLOGI DENGAN METODE  
*COLLABORATIVE FILTERING***

**ARITHA HANDRICO**

**10851001718**

Tanggal Sidang: 28 November 2012

Periode Wisuda: Februari 2013

Jurusan Teknik Informatika

Fakultas Sains dan Teknologi

Universitas Islam Negeri Sultan Syarif Kasim Riau

**ABSTRAK**

Perpustakaan merupakan sarana terpenting pada universitas. Semakin banyak koleksi buku pada perpustakaan akan semakin membantu kegiatan riset dan akademis mahasiswa. Namun karena banyaknya koleksi buku tersebut mahasiswa sering mendapat kesulitan menemukan buku yang sesuai dengan minat dan kebutuhannya. Pada penelitian ini, sistem rekomendasi buku perpustakaan Fakultas Sains dan Teknologi dengan menggunakan metode *collaborative filtering* akan memberikan rekomendasi buku kepada mahasiswa berdasarkan korelasi peminjaman dan pencarian yang pernah ia lakukan dengan yang dilakukan orang lain. Data pencarian dan peminjaman akan diubah menjadi *rating* kemudian dilakukan perhitungan dengan algoritma *Adjusted Cossine Similarity* dan *Weight Sum* sehingga menghasilkan prediksi *rating* terhadap buku yang belum pernah di-*rating* oleh mahasiswa. Sistem rekomendasi ini telah diuji dengan menggunakan metode *black box*, pengujian akurasi dengan MAE (*means absolute error*) dan pengujian waktu eksekusi. Hasil pengujian menunjukkan nilai MAE terendah didapatkan ketika tingkat *sparsity* sebesar 30% yaitu rata-rata 0.715677. Sistem dapat mengatasi tingkat *sparsity* hingga 60% dengan rata-rata MAE yang diperoleh sebesar 0.884307. Waktu eksekusi yang dibutuhkan dipengaruhi oleh jumlah *item* yang di-*rating* dan tingkat *sparsity*. Meningkatnya jumlah *item* yang di-*rating* diiringi dengan peningkatan waktu eksekusi yang dibutuhkan. Sedangkan untuk tingkat *sparsity* semakin tinggi tingkat *sparsity*-nya waktu eksekusi yang dibutuhkan relatif lebih sedikit.

**Kata kunci:** *Adjusted Cossine Similarity, Collaborative Filtering Recommender System, Means Absolute Error, Sistem Rekomendasi Buku, Weight Sum.*

## DAFTAR ISI

LEMBAR PERSETUJUAN.....	ii
LEMBAR PENGESAHAN .....	iii
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL.....	iv
LEMBAR PERNYATAAN .....	v
LEMBAR PERSEMBAHAN .....	vi
ABSTRAK .....	vii
<i>ABSTRACT</i> .....	viii
KATA PENGANTAR .....	ix
DAFTAR ISI .....	xi
DAFTAR GAMBAR .....	xiv
DAFTAR TABEL.....	xvi
DAFTAR RUMUS .....	xvii
DAFTAR ISTILAH .....	xviii
DAFTAR LAMPIRAN.....	xx
DAFTAR SIMBOL.....	xxi
BAB I PENDAHULUAN .....	I-1
1.1. Latar Belakang .....	I-1
1.2. Rumusan Masalah.....	I-3
1.3. Batasan Masalah .....	I-3
1.4. Tujuan .....	I-4
1.5. Sistematika Penulisan .....	I-4
BAB II LANDASAN TEORI .....	II-1
2.1. Sistem Rekomendasi.....	II-1
2.1.1. <i>Content Based Filtering</i> .....	II-2
2.1.2. <i>Collaborative Filtering</i> .....	II-2
2.1.2.1 Penemuan <i>Similar Item</i> .....	II-7
2.1.2.2 Perhitungan Prediksi.....	II-9
2.2. Skala Penilaian .....	II-11
2.2.1. <i>Skala Likert</i> .....	II-11

2.2.2. Skala Guttman .....	II-11
2.2.3. <i>Rating Scale</i> .....	II-12
2.3. Evaluasi Sistem Rekomendasi .....	II-13
2.3.1. Jenis-Jenis Perhitungan <i>Error</i> .....	II-13
2.3.2. Presisi ( <i>Precision</i> ).....	II-15
2.4. <i>Sparsity</i> .....	II-15
BAB III METODOLOGI PENELITIAN .....	III-1
3.1. Data Penelitian.....	III-1
3.2. Tahapan Penelitian .....	III-1
3.2.1. Pengumpulan Data.....	III-2
3.2.2. Analisa.....	III-2
3.2.3. Perancangan.....	III-3
3.2.4. Implementasi .....	III-4
3.2.5. Pengujian .....	III-4
3.2.6. Kesimpulan dan Saran .....	III-5
BAB IV ANALISA DAN PERANCANGAN.....	IV-1
4.1. Analisa Sistem Rekomendasi .....	IV-1
4.1.1. Analisa Permasalahan.....	IV-1
4.1.2. Analisa Kebutuhan Data.....	IV-5
4.1.3. Analisa Kebutuhan Perangkat Keras dan Perangkat Lunak.....	IV-6
4.1.4. Analisa Metode.....	IV-7
4.1.4.1. Pemberian <i>Rating</i> .....	IV-7
4.1.4.1. Perhitungan Kemiripan ( <i>Similarity</i> ).....	IV-8
4.1.4.2. Perhitungan Prediksi .....	IV-11
4.1.5. Analisa Fungsional Sistem .....	IV-13
4.1.5.1. <i>Context Diagram</i> .....	IV-134
4.1.5.2. <i>Data Flow Diagram</i> .....	IV-14
4.1.5.3. <i>Entity Relationship Diagram</i> .....	IV-15
4.2. Perancangan Sistem.....	IV-16
4.2.1. Perancangan Tabel.....	IV-16

4.2.2. Perancangan Antarmuka.....	IV-19
4.2.2.1. Tampilan Utama.....	IV-19
4.2.2.2. Tampilan Hasil Pencarian .....	IV-20
4.2.2.3. Tampilan Peminjaman .....	IV-20
4.2.2.4. Hasil Rekomendasi.....	IV-21
4.2.2.5. Halaman Utama Administrator .....	IV-21
4.2.2.6. Halaman Tampil Data Untuk Administrator. ....	IV-22
4.2.2.7. Halaman Tambah Data.....	IV-22
4.2.2.8. Halaman <i>Import Data Excel</i> .....	IV-23
<b>BAB V IMPLEMENTASI DAN PENGUJIAN .....</b>	<b>V-1</b>
5.1. Implementasi Sistem .....	V-1
5.1.1. Batasan Implementasi.....	V-1
5.1.2. Lingkungan Operasional .....	V-1
5.1.3. Hasil Implementasi .....	V-2
5.2. Pengujian .....	V-10
5.2.1. Pengujian <i>Black Box</i> pada Sistem Rekomendasi	
Buku Perpustakaan.....	V-10
5.2.2. Pengujian Kualitas Rekomendasi.....	V-13
5.2.2.1. Prosedur Pengujian Kualitas Rekomendasi ....	V-13
5.2.2.2. Hasil Pengujian Kualitas Rekomendasi .....	V-14
5.2.3. Pengujian Waktu Eksekusi .....	V-18
5.2.3.1. Prosedur Pengujian Waktu Eksekusi .....	V-19
5.2.3.2. Hasil Pengujian Waktu Eksekusi .....	V-19
5.2.4. Analisa Hasil Pengujian .....	V-20
5.2.5. Kesimpulan Pengujian.....	V-22
<b>BAB VI PENUTUP .....</b>	<b>VI-1</b>
6.1. Kesimpulan.....	VI-1
6.2. Saran .....	VI-2
<b>DAFTAR PUSTAKA .....</b>	<b>xii</b>
<b>LAMPIRAN</b>	
<b>RIWAYAT HIDUP</b>	

## DAFTAR TABEL

Tabel	Halaman
4.1 Nilai <i>Rating User</i> Terhadap <i>Item</i> .....	IV-9
4.2 Nilai kemiripan antar <i>item</i> .....	IV-11
4.3 Nilai kemiripan untuk <i>user</i> Riesti .....	IV-11
4.4 <i>Context Diagram</i> Sistem Rekomendasi Buku Perpustakaan .....	IV-13
4.5 DFD <i>level</i> 1 Sistem Rekomendasi .....	IV-15
4.6 Administrator .....	IV-16
4.7 Mahasiswa.....	IV-17
4.8 Buku .....	IV-17
4.9 Peminjaman.....	IV-18
4.10 Pencarian.....	IV-18
4.11 <i>Rating</i> .....	IV-18
4.12 <i>Similarity</i> .....	IV-18
4.13 Prediksi .....	IV-19
5.1 Hasil Pengujian Sistem dengan Metode <i>Blackbox</i> .....	V-11
5.2 Jawaban Hasil Pengujian dengan Kuisioner Kepada Mahasiswa .....	V-15
5.3 Nilai MAE Berdasarkan Tingkat <i>Sparsity</i> .....	V-18
5.4 Waktu Eksekusi Berdasarkan Jumlah <i>Item</i> .....	V-19
5.5 Waktu Eksekusi Berdasarkan Tingkat <i>Sparsity</i> .....	V-20
5.6 Perbandingan Rata-Rata MAE dan Waktu Eksekusi.....	V-22
A.1 DFD <i>Level</i> 2 Proses 2 Pengolahan Data Master .....	A-1
A.2 DFD <i>Level</i> 2 Proses 2 Pemberian <i>Rating</i> .....	A-2
A-3 DFD <i>Level</i> 2 Proses 5 Pemberian Rekomendasi.....	A-3
A.4 DFD <i>Level</i> 3 Proses 2.1 Pengolahan Data Admin .....	A-4
A.5 DFD <i>Level</i> 3 Proses 2.2 Pengolahan Data Buku.....	A-5
A.6 DFD <i>Level</i> 3 Proses 2.1 Pengolahan Data Mahasiswa .....	A-6

## DAFTAR GAMBAR

Gambar	Halaman
2.1 Contoh Sistem Rekomendasi Pada Youtube.com .....	II-2
2.2 Proses <i>Collaborative Filtering</i> (Hakim,2010) .....	II-3
2.3 Skema <i>Item-based Collaborative Filtering</i> (Hakim,2010) .....	II-6
3.1 Tahapan Penelitian .....	III-2
4.1 Deskripsi Umum Sistem .....	IV-2
4.2 <i>Flowchart</i> Sistem Rekomendasi Buku Perpustakaan .....	IV-4
4.3 <i>Context Diagram</i> Sistem Rekomendasi Buku Perpustakaan .....	IV-13
4.4 DFD <i>level</i> 1 Sistem Rekomendasi Buku Perpustakaan.....	IV-14
4.5 ERD Sistem Rekomendasi .....	IV-16
4.6 Antarmuka <i>Form</i> Utama .....	IV-19
4.7 Antarmuka Hasil Pencarian.....	IV-20
4.8 Antarmuka Peminjaman .....	IV-20
4.9 Antarmuka Hasil Rekomendasi.....	IV-21
4.10 Antarmuka Halaman Administrator.....	IV-21
4.11 Antarmuka Tampil Data.....	IV-22
4.12 Antarmuka Tambah Data .....	IV-22
4.13 Antarmuka Import Data Excel .....	IV-23
5.1 Tampilan Menu <i>Login</i> .....	V-3
5.2 Tampilan Menu <i>Home</i> .....	V-3
5.3 Hasil Tampilan Hasil Pencarian.....	V-4
5.4 Tampilan Menu Detail Buku.....	V-4
5.5 Tampilan Menu Buku Rekomendasi.....	V-5
5.6 Tampilan Menu Buku Terbaru.....	V-5
5.7 Tampilan Menu Petunjuk.....	V-6
5.8 Tampilan Menu Tentang Kami .....	V-6
5.9 Tampilan Halaman Admin.....	V-7
5.10 Tampilan Halaman Tambah Data .....	V-7

5.11	Tampilan Halaman <i>Import Data excel</i> .....	V-8
5.12	Konversi Peminjaman ke <i>Rating</i> .....	V-9
5.13	Konversi Pencarian ke <i>Rating</i> .....	V-9
A.1	DFD <i>Level 2</i> Proses 2 Pengolahan Data Master .....	A-1
A.2	DFD <i>Level 2</i> Proses 3 Pemberian <i>Rating</i> .....	A-2
A-3	DFD <i>Level 2</i> Proses 5 Pemberian Rekomendasi.....	A-3
A.4	<i>Level 3</i> Proses 2.1 Pengolahan Data Admin .....	A-4
A.5	DFD <i>Level 3</i> Proses 2.2 Pengolahan Data Buku.....	A-5
A.6	DFD <i>Level 3</i> Proses 2.3 Pengolahan Data Mahasiswa .....	A-5



## DAFTAR RUMUS

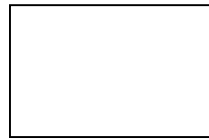
<b>Rumus</b>	<b>Halaman</b>
2.1. <i>Cosine-based similarity</i> .....	II-7
2.2. <i>Correlation-based similarity</i> .....	II-8
2.3. <i>Adjusted Cosine Similarity</i> .....	II-8
2.4. <i>Regression</i> .....	II-9
2.5. <i>Weighted Sum</i> .....	II-10
2.6. <i>Mean Square Error</i> .....	II-13
2.7. <i>Mean Absolute Percentage</i> .....	II-14
2.8. <i>Mean Absolute Error</i> .....	II-14
2.9. <i>Precision</i> .....	II-14

## DAFTAR ISTILAH

<b><i>Collabaraotive Filtering</i></b>	= Teknik dalam <i>recommender system</i> yang dalam menentukan preferensi <i>user</i> berdasarkan keterkaitan antara <i>user active</i> dengan <i>user</i> lainnya
<b><i>Context Diagram</i></b>	= Gambaran umum dari sistem yang akan dibangun
<b><i>Database</i></b>	= Basis data yang berisi kumpulan data-data hasil pengamatan
<b><i>Data Flow Diagram</i></b>	= Menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan
<b><i>Dataset</i></b>	= objek yang merepresentasikan data dan relasinya di <i>memory</i>
<b><i>Eksplisit</i></b>	= Secara sadar
<b><i>Entity Relationship Diagram</i></b>	= Objek data dan hubungan antar diagram
<b><i>Form</i></b>	= Bentuk dari sebuah tampilan
<b><i>Implementasi</i></b>	= Pelaksanaan atau penerapan
<b><i>Implisit</i></b>	= Secara tidak sadar
<b><i>Input</i></b>	= Data yang dimasukkan
<b><i>Interface</i></b>	= Tampilan antar muka
<b><i>Item</i></b>	= Benda yang akan direkomendasikan, pada penelitian ini yaitu buku-buku perpustakaan
<b><i>Korelasi</i></b>	= Saling berkaitan/berhubungan
<b><i>Output</i></b>	= Data yang dihasilkan
<b><i>Poor Accuracy</i></b>	= nilai akurasi yang rendah
<b><i>Preferensi</i></b>	= Lebih disukainya suatu alternatif
<b><i>Similarity</i></b>	= Nilai yang digunakan untuk mencari kemiripan antar dua <i>item</i>
<b><i>Smoothing</i></b>	= Proses penghalusan item-item yang belum di- <i>rating</i> dengan memberikan <i>rating</i> bayangan

<b><i>Sparsity</i></b>	= Keadaan <i>rating</i> matriks yang jarang
<b><i>Sparse</i></b>	= Jarang/renggang
<b><i>Survey</i></b>	= Penelitian, peninjauan atau penyelidikan
<b><i>Recommender System</i></b>	= Sistem yang membantu user memberikan rekomendasi atau saran dari sekian banyak item
<b><i>Testing</i></b>	= Pengujian (percobaan) untuk mengetahui tingkat kemampuan atau mengetahui mutunya
<b><i>User</i></b>	= Pemakai sistem (mahasiswa)
<b><i>User Active</i></b>	= <i>User</i> yang akan diberikan prediksi <i>rating</i>

## DAFTAR SIMBOL



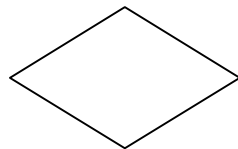
Proses pada *flowchart*



*Start/Finish* suatu proses pada *flowchart*



Alur/ langkah pada *flowchart* dan model data spasial



Kondisi/keadaan tindakan pada *flowchart* berupa Ya atau Tidak



Basisdata pada *flowchart*



Input/output pada *flowchart*



*Datastore*



*Display*

## DAFTAR LAMPIRAN

Lampiran	Halaman
A. Rincian <i>Data Flow Diagram</i> .....	A-1
B. Form Kuisisioner Penelitian Tugas Akhir .....	B-1

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Perpustakaan merupakan elemen yang sangat penting dalam sebuah Perguruan Tinggi/Universitas. Dengan adanya perpustakaan, proses pembelajaran mahasiswa tidak terpaku hanya pada materi yang diajarkan pendidik di kelas. Mahasiswa dapat lebih mendalami materi pelajaran dengan mencari referensi berupa buku, laporan kerja praktek dan tugas akhir di perpustakaan.

Perpustakaan perguruan tinggi harus mempunyai filosofi “*student first*” bagaimana mahasiswa mendapatkan pelayanan yang terbaik dan tepat guna sehingga kegiatan riset dan akademis lainnya mampu terpenuhi dengan baik. Perpustakaan perguruan tinggi dituntut untuk lebih *go public* dan *accessible* sehingga dapat menarik minat baca mahasiswa dan mendukung kegiatan keilmiah kampus.

Selain sebagai sumber ilmu pengetahuan, perpustakaan juga menjadi patokan apakah sebuah universitas dapat dikatakan universitas yang bertaraf internasional atau tidak, karena salah satu indikator sebuah “*World Class University*” adalah universitas yang memiliki perpustakaan yang bertaraf internasional (<http://pustaka.uns.ac.id>). Sebuah perpustakaan yang bertaraf internasional sudah pasti memiliki koleksi buku yang beragam sehingga dapat digunakan mahasiswa sebagai referensi. Namun karena keberagaman koleksi buku tersebut, terkadang mahasiswa mengalami kesusahan untuk mencari buku yang tepat, yang sesuai dengan yang mereka butuhkan.

Dengan permasalahan di atas tentunya mahasiswa akan terbantu jika ada dosen atau orang lain merekomendasikan sebuah buku yang sesuai dengan topik yang mereka inginkan. Tapi tidak mungkin setiap saat ada orang yang dapat merekomendasikan buku-buku tersebut kepada mahasiswa, maka dari itu sebuah

sistem rekomendasi (*recommender system*) yang dapat memberikan saran atau rekomendasi buku yang sesuai berdasarkan ketertarikan mahasiswa dan kriteria buku akan dapat membantu dalam pencarian referensi untuk mahasiswa.

Pendekatan pada *Recommender System* (RS) dalam proses rekomendasi terdiri atas dua cara, yaitu *content based filtering* dan *collaborative filtering*. *Content based filtering* memberikan rekomendasi berdasarkan deskripsi dari *item*. Metode ini memiliki kekurangan yaitu ketika jumlah *item* yang akan direkomendasikan menjadi bertambah besar, sistem akan membutuhkan waktu yang lama untuk melakukan ekstraksi pada setiap *item* yang akan direkomendasikan. (Pazzani, 2007) pada skripsi (Hakim, 2010). Berbeda dengan metode *Content-based filtering*, metode *Collaborative filtering* menghasilkan rekomendasi berdasarkan keterkaitan antar *user* yang menyukai suatu *item* tertentu. Pada *collaborative filtering*, rekomendasi berdasarkan pada keterkaitan yang didapat diantara *user* yang telah *me-rating* atau bertransaksi dalam sistem. Misalnya *user* A menyukai *item* X dan Y, kemudian *user* B menyukai *item* Y maka sistem akan merekomendasikan *item* X kepada *user* B.

Penelitian tentang *Recommender System* (RS) buku perpustakaan pernah dilakukan oleh Adi Wibowo, Andreas Handojo dan Minardi Taliwang pada penelitian yang berjudul “*Recommender System* di Perpustakaan Universitas Kristen Petra menggunakan *Rocchio Relevance Feedback* dan *Cosine Similarity*”. Penelitian tersebut menggunakan metode *content-based filtering* dimana metode tersebut mempunyai kelemahan yaitu ketika jumlah *item* yang direkomendasi bertambah besar, sistem akan membutuhkan waktu lebih untuk mengekstraksi informasi yang ada pada setiap *item*. Maka dari itu, untuk penelitian selanjutnya penulis ingin menerapkan metode *collaborative filtering* untuk merekomendasikan buku-buku perpustakaan Fakultas Sains dan Teknologi Universitas Islam Sultan Syarif Kasim Riau.

Penelitian sistem rekomendasi dengan menggunakan metode *collaborative filtering* pernah diterapkan pada sistem rekomendasi pembelian buku secara online oleh Shofwatul ‘Uyun. Sistem rekomendasi pada toko buku *online* berhasil menampilkan rekomendasi buku kepada pelanggan sesuai dengan *rating* yang diberikan oleh pelanggan terhadap buku yang diminati pelanggan tersebut.

Penelitian lain mengenai metode *collaborative filtering* juga pernah dilakukan oleh Badrul Sarwar dimana pada penelitian ini dibandingkan algoritma yang dapat digunakan untuk metode *collaborative filtering*. Hasil dari penelitian tersebut algoritma yang bagus untuk *collaborative filtering* adalah algoritma *adjusted cosine similarity* karena menghasilkan nilai *Mean Absolute Error* (MAE) yang paling kecil.

Berdasarkan latar belakang yang telah dipaparkan diatas, dalam Tugas Akhir ini penulis mencoba merancang bangun sebuah sistem rekomendasi buku perpustakaan yang dapat memberikan rekomendasi buku–buku yang diminati mahasiswa dengan metode *collaborative filtering*. Dengan demikian diharapkan mahasiswa dapat terbantu dalam menemukan referensi buku yang sesuai dengan kebutuhan mereka.

## **1.2. Rumusan Masalah**

Berdasarkan latar belakang yang telah diuraikan di atas, maka dapat ditarik sebuah rumusan masalah yang akan dijelaskan lebih lanjut pada laporan tugas akhir ini, yaitu “Bagaimana merancang bangun sebuah sistem rekomendasi buku perpustakaan dengan metode *collaborative filtering* agar dapat membantu mahasiswa merekomendasi buku yang sesuai dengan kebutuhan mahasiswa”.

## **1.3. Batasan Masalah**

Dalam pembuatan tugas akhir ini, diberi beberapa batasan masalah, yaitu:

1. *Rating user* dalam pereferensi pengguna digantikan dengan melihat *history* peminjaman dan pencarian yang dilakukan pengguna, dengan asumsi ketika *user* meminjam buku perpustakaan berarti *user* telah memberikan *rating* pada buku tersebut.
2. Hanya menggunakan teknik *item based collaborative filtering* karena pada teknik ini *item*/buku yang pernah dipinjam atau dicari *user* akan menjadi patokan untuk pemberian rekomendasi.
3. Algoritma yang digunakan pada tahap penemuan kemiripan adalah *adjusted cosine similarity* dan pada tahap penghitungan prediksi



menggunakan algoritma *weight sum* karena kedua algoritma ini yang menghasilkan nilai MAE terendah (Sarwar, 2001).

4. Pencarian buku hanya menggunakan *query sql* karena fokus penelitian pada metode *collaborative filtering* untuk rekomendasi buku perpustakaan Fakultas Sains dan Teknologi.

#### **1.4. Tujuan Penelitian**

Tujuan yang ingin dicapai pada penulisan tugas akhir ini adalah, “Menghasilkan sebuah sistem rekomendasi buku di perpustakaan Fakultas Sains dan Teknologi dengan menggunakan metode *collaborative filtering*”.

#### **1.5. Sistematika Penulisan**

Dalam penyusunan laporan tugas akhir ini, sistematika penulisan yang penulis gunakan adalah sebagai berikut :

##### **1. Bab I Pendahuluan**

Bab I ini merupakan bagian yang akan menguraikan hal-hal seperti; latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, dan sistematika penulisan laporan tugas akhir.

##### **2. Bab II Landasan Teori**

Bab ini berisi tentang teori-teori tentang *Recomendation System* (sistem rekomendasi) dan metode *Content Based Filtering* yang digunakan pada sistem rekomendasi.

##### **3. Bab III Metodologi Penelitian**

Bab ini berisi tentang cara-cara atau hal-hal yang dilakukan dalam menyelesaikan kasus tugas akhir ini.

##### **4. Bab IV Analisa dan Perancangan**

Bab ini berisi tentang analisa dari penelitian yang dilakukan dalam tugas akhir ini sekaligus menerangkan perancangan sistem rekomendasi yang dibangun.

## **5. Bab V Implementasi dan Pengujian**

Bab ini berisi tentang langkah-sistem rekomendasi buku perpustakaan dan menguji hasil dari rancangan yang telah dibangun.

## **6. Bab VI Penutup**

Bab ini berisi kesimpulan dan saran mengenai hasil analisa, perancangan, hasil implementasi dan hasil pengujian yang telah dilakukan terhadap rancang bangun sistem rekomendasi yang telah dibangun.

## BAB II

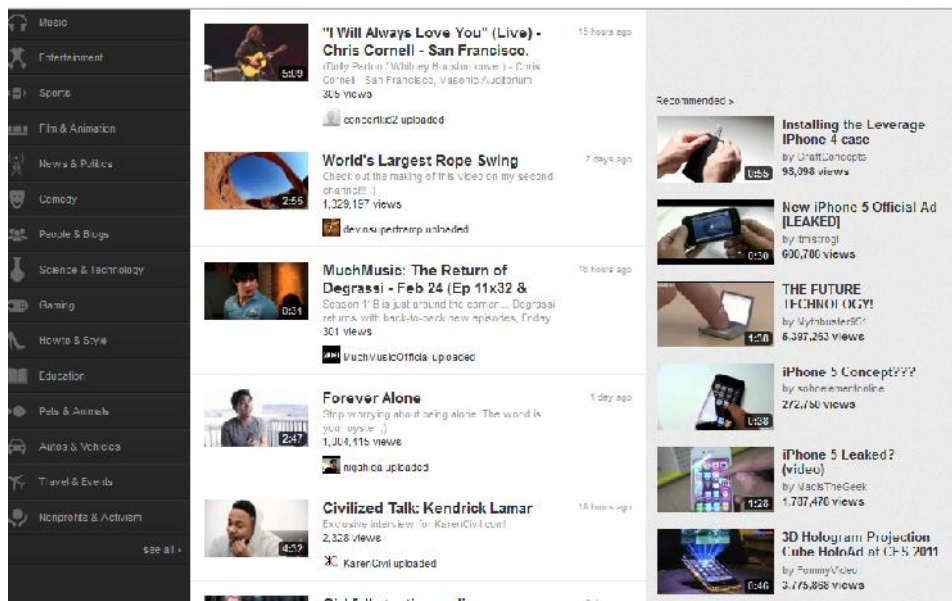
### LANDASAN TEORI

#### 2.1. Sistem Rekomendasi

Sistem rekomendasi merupakan suatu sistem yang dirancang untuk memprediksi sekumpulan item yang sesuai dengan preferensi *user* yang mana nantinya item tersebut akan direkomendasikan pada user (Sanjung,2011). Perkiraan informasi ini diperoleh melalui profil pengguna (*user*), preferensi *item* dan aktivitas yang terjadi pada sistem. Profil pengguna dapat berisi tentang informasi pengguna (*preferensi user*), ketertarikan pengguna pada suatu *item* dan juga *history* interaksi antara pengguna dengan *item*. Misalnya ketika pengguna melakukan peminjaman buku maka data *history* peminjaman ini akan dicatat dan menjadi profil pengguna.

Pada sistem rekomendasi terdapat sekumpulan *item* yang ada akan disaring (*filtering*) berdasarkan preferensi *item* preferensi *user*, *rating*, catatan *history* transaksi dan lain lain sehingga menghasilkan beberapa *item* yang direkomendasikan kepada *user*. Pada saat ini sudah banyak yang menerapkan sistem rekomendasi pada website untuk mempermudah *user* mencari informasi seperti youtube.com, amazon.com dan lain lain.

Gambar 2.1 merupakan *screenshot* dari halaman situs youtube.com. Pada halaman tersebut dapat dilihat pada sisi sebelah kanan halaman youtube terdapat beberapa video yang direkomendasikan sistem untuk *user*. Video-video tersebut direkomendasikan kepada *user* berdasarkan *history* pencarian yang pernah dilakukan *user* pada situs youtube.com.



Gambar 2.1 Contoh Sistem Rekomendasi Pada Youtube.com

Secara umum terdapat 2 metode untuk membangun sebuah sistem rekomendasi yaitu, *content based filtering* dan *collaborative filtering*.

### 2.1.1. *Content Based Filtering*

*Content based filtering* memberikan rekomendasi suatu *item* untuk seorang *user* berdasarkan deskripsi dari *item* yang akan direkomendasikan tersebut serta profil dari ketertarikan seorang *user*. Sistem rekomendasi *content based* menganalisa deskripsi dari setiap *item* untuk mengidentifikasi *item* mana yang mempunyai ketertarikan khusus terhadap *item* yang disukai *user*. (Pazzani, 2007)

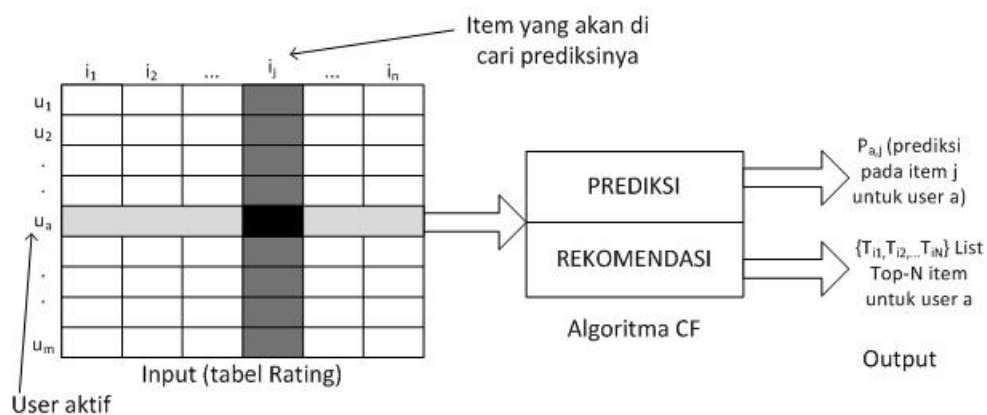
Sistem rekomendasi dengan metode *content based filtering* biasa digunakan untuk merekomendasikan berita, artikel maupun situs web. Metode tersebut akan mengekstrak informasi yang terdapat pada *item* kemudian membandingkannya dengan informasi *item* yang pernah dilihat atau disukai oleh *user*.

### 2.1.2. *Collaborative Filtering*

*Collaborative Filtering* merupakan salah satu cara yang diterapkan sistem rekomendasi untuk memberikan prediksi otomatis terhadap keinginan pelanggan dengan cara mengoleksi informasi dari banyak pelanggan (Ampaziz, 2008). Pada *collaborative filtering*, rekomendasi yang diberikan berdasarkan

keterkaitan(preferensi) antaraitem satu dengan item lainnya dan keterkaitan yang didapat diantara useryang telah me-rating atau berinteraksi dalam sistem.

*Collaborative filtering* memberikan rekomendasi berdasarkan kumpulan dari pendapat, minat dan ketertarikan beberapa useryang biasanya diberikan dalam bentuk *rating* yang diberikan userkepada suatu item. Pada metode *Collaborative filtering*, terdapat kumpulan  $m$  user  $U = \{u_1, u_2, u_3, \dots, u_m\}$  dan kumpulan  $n$  item  $I = \{i_1, i_2, i_3, \dots, i_n\}$  dimana setiap user ( $u_i$ ) mempunyai daftar item  $I_{u_i}$  yang merupakan ekpresi dari pendapatnya.



Gambar 2.2 Proses *Collaborative Filtering* (Hakim,2010)

Gambar 2.2 diatas merupakan diagram skema dari metode *collaborative filtering*. Pada gambar tersebut direpresentasikan sejumlah  $m$  user x  $n$  item sebagai sebuah matriks *rating* dimana berisi nilai *rating* dari user untuk setiap item.

User aktif ( $U_a$ ) pada skema tersebut adalah user yang akan diberikan rekomendasi item yang mungkin disukainya. Item tersebut direpresentasikan dalam dua bentuk (Ampazis,2012) yaitu :

1. Prediksi : merupakan nilai numerik, dimana  $P_{a,j}$  adalah nilai prediksi *rating* item yang mungkin disukai oleh  $U_a$ .
2. Rekomendasi : merupakan daftar  $N$  item yang mungkin disukai oleh  $U_a$ . Dengan catatan bahwa item yang direkomendasikan belum pernah dibeli atau dirating oleh  $U_a$ . Hal ini sering juga disebut *top-N recommendation*.

Untuk memperoleh data *rating* dari *user* yang digunakan dalam sistem rekomendasi, dibedakan menjadi dua cara (Wibowo, 2010) yaitu:

1. Secara Eksplisit, yaitu proses pengumpulan data dimana *user* memberikan data secara sadar/sengaja contohnya : pemberian *rating*, pengisian form tentang data *user*, komentar *user* pada suatu *item*.
2. Secara Implisit, yaitu proses pengumpulan data dimana *user* tidak menyadari bahwa ia telah memberikan masukan terhadap sistem. Contohnya : catatan *item* yang dibeli atau dipinjam pada perpustakaan, *item* yang diklik, *item* yang dicari *user* dan lain lain.

Umumnya pada *collaborative filtering*, untuk menunjukkan bahwa *user* memiliki ketertarikan terhadap suatu *item* yaitu dengan pemberian *rating* secara eksplisit dengan member nilai terhadap *item* tersebut. Namun, pemberian *rating* dengan cara ini seringkali merepotkan *user* karena harus me-*rating* *item* satu persatu.

Cara lain yang lebih cerdas yaitu dengan menggunakan metode implisit untuk memperoleh nilai *rating* dari *user* (Claypool, 2001). Dimana *rating* diperoleh secara langsung ketika *user* berinteraksi dengan sistem. *User* tidak menyadari bahwa aktifitas yang mereka lakukan seperti mencari, melakukan klik, melihat suatu *item* dicatat sebagai nilai *rating*.

Pengambilan *rating* secara implisit dapat mengumpulkan data *rating* tanpa harus menugaskan *user* untuk member penilaian. Berdasarkan penelitian yang dilakukan groupLens System, *user* akan lebih banyak membaca artikel dari pada memberikan *rating* pada suatu artikel (Claypool, 2001). Hal ini terjadi karena tujuan *user* yang sebenarnya ketika membuka halaman artikel adalah untuk membacanya bukan untuk memberikan *rating*. Dengan banyaknya *rating* yang terkumpul melalui aktifitas *user* akan sangat berguna pada *collaborative filtering* karena metode ini membutuhkan banyak *rating* yang harus diinputkan agar hasil rekomendasi yang ditawarkan berkualitas.

Banyak cara untuk memperoleh penilaian *user* secara implisit, berikut ini beberapa cara yang dapat dilakukan untuk memperoleh nilai *rating* secara implisit:

- a. *Time on page*, sistem akan menghitung berapa lama *user* membuka suatu halaman. Perhitungan dimulai ketika halaman selesai dibuka sampai halaman ditutup atau *user* berpindah ke halaman lain.
- b. *Time moving mouse*, sistem menghitung berapa banyak kursor *mouse* berpindah dari posisi awalnya. Beberapa *user* memindahkan posisi *mouse* ketika membaca informasi, melihat objek yang menarik atau melihat link yang menarik. Semakin banyak pergerakan *mouse* yang dilakukan ketertarikan *user* pun semakin besar.
- c. *Number of mouse click, mouseclick* dapat dijadikan indikator ketertarikan *user*. Setiap klik yang dilakukan *user* terhadap suatu gambar, objek maupun link menyatakan bahwa *user* tertarik kepada *item* tersebut. Jumlah klik yang dilakukan dicatat sistem sebagai suatu penilaian *user*.
- d. *Save reference or save an object, user* yang merasa bahwa informasi yang terkandung pada suatu referensi atau objek merupakan informasi yang penting sering melakukan penyimpanan referensi atau objek tersebut sehingga dapat dibuka kembali ketika dibutuhkan. Setiap penyimpanan dilakukan menggambarkan ketertarikan *user* terhadap *item* tersebut. Semakin banyak penyimpanan objek yang dilakukan maka semakin tinggi pula nilai ketertarikan *user* terhadap *item* tersebut.

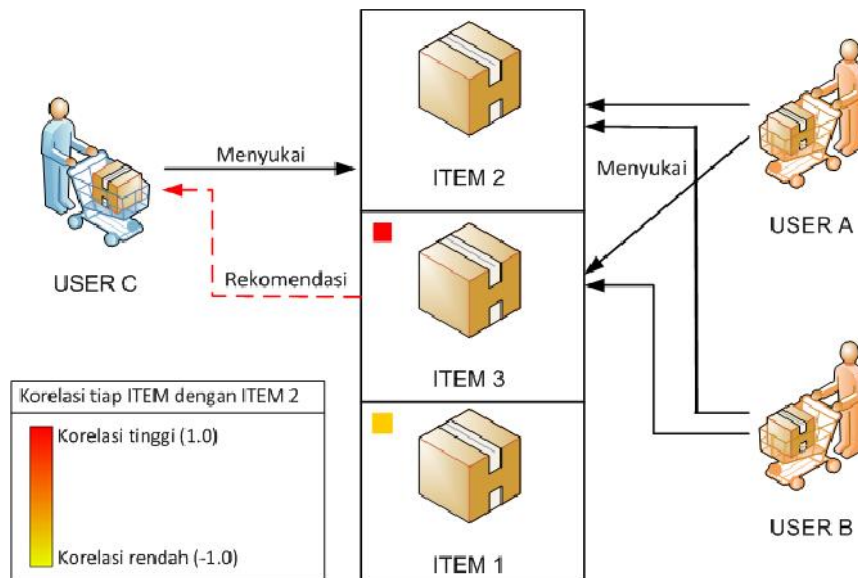
Banyak cara lain untuk mengumpulkan penilaian *user* terhadap suatu *item*, objek atau artikel misalnya menghitung kata yang diblock *user*, *item* yang dibeli dan lain lain. Inti dari penilaian secara implisit yaitu bagaimana agar *user* memberikan penilaian yang sebanyak banyaknya tetapi *user* tidak menyadari bahwa ia telah melakukan penilaian.

Pendekatan *collaborative filtering* pada dasarnya dibagi menjadi dua kategori yaitu *userbased collaborative filtering* disebut juga *memorybased*, dan *item based collaborative filtering* yang disebut juga *model-based* (Uyun, 2011).

Pada pendekatan *user based collaborative filtering* sistem memberikan rekomendasi kepada *user* *item-item* yang disukai atau dirating oleh *user-user* lain yang memiliki banyak kemiripan dengannya. Misalnya, *user a* menyukai atau merating *item1 2* dan *3*, kemudian *userb* menyukai *item1 2* dan *4* maka sistem

akan merekomendasikan *item3* kepada *userb* dan *item 4* kepada *usera*. Kelebihan dari pendekatan *user based collaborative filtering* adalah dapat menghasilkan rekomendasi yang berkualitas baik. Sedangkan kekurangannya adalah kompleksitas perhitungan akan semakin bertambah seiring dengan bertambahnya pengguna sistem, semakin banyak pengguna (*user*) yang menggunakan sistem maka proses perekomendasi akan semakin lama.

Pendekatan *item based collaborative filtering* memberikan rekomendasi berdasarkan kemiripan antar *item*. Metode ini merupakan metode rekomendasi yang didasari atas adanya kesamaan antara pemberian *rating* terhadap suatu *item* dengan *item* yang pernah di-*rating* user lain (Purwanto, 2009). *Item* yang telah di-*rating* oleh *user* akan menjadi patokan untuk mencari sejumlah *item* lainnya yang berkorelasi dengan *item* yang telah di-*rating* user. Motivasi kunci dibalik metode ini adalah *user* akan cenderung menyukai *item* yang sejenis atau mempunyai korelasi dengan *item* yang telah disukainya (Sarwar, 2001).



Gambar 2.3 Skema *Item-based Collaborative Filtering* (Hakim, 2010)

Gambar 2.3 menggambarkan bagaimana *item 3* direkomendasikan kepada *user c* yang telah menyukai *item 2*, dimana *item2* yang disukai *userc* memiliki korelasi yang kuat dengan *item 3*.



Secara umum proses pemberian rekomendasi pada *collaborative filtering* terdiri atas 2 langkah (Sarwar, 2001), yaitu:

1. Penemuan *similar item*,
2. Penghitungan prediksi.

#### 2.1.2.1. Penemuan *Similar Item*

Tahap ini merupakan tahap untuk mencari *item-item* yang mirip. Setiap *item* yang pernah disukai *user* akan dibandingkan dengan *item* lain yang belum pernah disukai *user*. Hasil dari tahapan ini adalah nilai *similarity* (kemiripan) yang menggambarkan seberapa mirip suatu *item* dengan *item* lain. Terdapat beberapa algoritma untuk menemukan *similar item*, yaitu: (Sarwar, 2001)

##### a. Algoritma *Cosine-based Similarity*

Pada kasus ini dua *item* dianggap sebagai 2 vektor. Kesamaan antara 2 *item* ini diukur dengan menghitung kosinus dari sudut antara 2 vektor *item*.

*Item* dibandingkan misalnya  $u$  dan  $v$ , dianggap sebagai sebuah vektor baris dengan anggotanya adalah nilai *rating* yang diberikan terhadap kedua *item* tersebut. Dua vektor dikatakan sama jika membentuk sudut  $0^\circ$  atau nilai kosinusnya 1. Dengan kata lain dua *item* dikatakan mirip jika nilai kosinus dari perhitungan mendekati 1 (Wiranto, 2010).

Persamaan *cosine-based similarity*:

$$sim(i, j) = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} * \vec{j}}{||\vec{i}||_2 * ||\vec{j}||_2} \quad \dots(2.1)$$

Dimana  $\vec{i}$  dan  $\vec{j}$  merupakan vektor vektor baris dengan anggota nilai *rating* pada *item*  $i$  dan *item*  $j$ .  $\cos(\vec{i}, \vec{j})$  merupakan nilai cosinus sudut yang dibentuk vektor baris *rating*  $i$  dan  $j$ .

##### b. Algoritma *Correlation-based Similarity*

Pada algoritma ini kemiripan antara dua *item*  $i$  dan  $j$  diukur dengan menghitung korelasi *Pearson-r correlation*. Agar perhitungan korelasi yang

diperoleh akurat, terlebih dahulu dilakukan pemisahan terhadap *co-rated items* (*item-item* yang kedua *item i* dan *j* nya di-*rating* oleh *user*).

Persamaan *correlation-based similarity*:

$$sim(i, j) = \frac{\sum_{u \in U} (R_{u,i} - \bar{R}_i)(R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R}_i)^2} \sqrt{\sum_{u \in U} (R_{u,j} - \bar{R}_j)^2}} \quad \dots(2.2)$$

c. *Adjusted-cossine similarity*

Persamaan *adjusted cosine similarity* digunakan untuk menghitung nilai kemiripan antar *item*. Perhitungan kemiripan ini merupakan modifikasi dari perhitungan kemiripan berbasis vektor dimana dengan melihat fakta bahwa setiap *user* memiliki skema *rating* yang berbeda-beda. Terkadang *user* memberi *rating* yang tinggi terhadap *item a* disisi lain *user* memberi *rating* yang sangat rendah pada *item b*. Maka dari itu untuk setiap *rating* dikurangi dengan rata-rata *rating* yang diberikan *user*.

Berdasarkan hasil penelitian (Sarwar, 2001) yang meneliti tentang algoritma-algoritma yang dapat digunakan untuk menghitung kemiripan, algoritma *adjusted-cossine similarity* merupakan algoritma yang dapat menghasilkan nilai MAE (*mean absolute error*) paling rendah dan dengan waktu yang paling cepat.

Persamaan *adjusted cosine* :

$$sim(i, j) = \frac{\sum_{u \in U} (R_{u,i} - \bar{R}_u)(R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R}_u)^2} \sqrt{\sum_{u \in U} (R_{u,j} - \bar{R}_u)^2}} \quad \dots(2.3)$$

Keterangan :

$Sim(i, j)$  = Nilai kemiripan antara *item i* dan *item j*

$u \in U$  = Himpunan *user* yang merating *item i* dan *j*

$R_{u,i}$  = *Rating user u* pada *item i*

$R_{u,j}$  = *Rating user u* pada *item j*

$\bar{R}_u$  = Nilai rata-rata *rating user u*

Untuk menghitung nilai kemiripan (*similarity*) antar 2 *item*, diperlukan himpunan *user* yang me-*rating* *item* tersebut. Nilai yang dihasilkan pada persamaan *adjusted-cosine similarity* adalah berkisar antara +1.0 dengan -1.0. *Item* dianggap saling berkolerasi jika nilai *similarity* antara kedua *item* tersebut mendekati +1, begitu juga sebaliknya *item* dianggap tidak berkolerasi apabila nilai *similarity*-nya mendekati -1.

#### 2.1.2.2. Perhitungan Prediksi

Setelah mendapatkan nilai kemiripan, maka langkah selanjutnya adalah proses perhitungan prediksi. Proses prediksi yang dilakukan adalah dengan memperkirakan nilai *rating* dari *user* terhadap suatu *item* yang belum pernah di-*rating* sebelumnya oleh *user* tersebut.

Algoritma yang dapat digunakan untuk mendapatkan prediksi dari suatu *item* yaitu: (Sarwar, 2001)

##### a. Algoritma *Regression*

Pendekatan ini mirip dengan metode *weighted sum* tapi metode ini tidak langsung menggunakan *rating* dari *item* yang mirip melainkan menggunakan penaksiran *rating* berdasarkan model regresi. Model ini memperkirakan nilai prediksi *rating* dari  $R'_N$  berdasarkan *linear regression model*.

Persamaan *regression*:

$$\bar{R}'_N = \alpha \bar{R}_i + \beta + \epsilon \quad \dots(2,4)$$

Parameter regresi  $\alpha$  dan  $\beta$  ditentukan berdasarkan kedua *rating vector* dan merupakan *error* dari model regresi.

##### b. Algoritma *weighted sum*

Algoritma ini mendapatkan nilai prediksi dengan menghitung total *rating* yang diberikan terhadap *item* yang mirip dengan *item* yang ingin diprediksi. Teknik ini memprediksi *item*  $j$  untuk *user* dengan menghitung jumlah

*rating* yang diberikan oleh *user* terhadap *item* yang berkorelasi dengan *item<sub>j</sub>*. Setiap *rating* yang diberikan *user* pada *item* yang berkorelasi dengan *item<sub>j</sub>* akan dikalikan dengan nilai kemiripannya. Kemudian dibagi dengan jumlah nilai absolut kemiripan seluruh *item* yang berkorelasi.

Persamaan *weighted sum* :

$$P(u, j) = \frac{\sum_{i \in I} (R_{u,i} * S_{i,j})}{\sum_{i \in I} |S_{i,j}|} \quad \dots(2.5)$$

Keterangan :

$P(u, j)$  = Prediksi untuk *user* pada *item<sub>j</sub>*

$i \in I$  = Himpunan *item* yang mirip dengan *item<sub>j</sub>*

$R_{u,i}$  = *Rating* *user* *u* pada *item<sub>i</sub>*

$S_{i,j}$  = Nilai kemiripan antar *item<sub>i</sub>* dan *item<sub>j</sub>*

Prediksi dengan persamaan *weighted sum* diperoleh dengan menghitung total keseluruhan dari perkalian antara *rating* *user* terhadap suatu *item* dan nilai *similarity* *item* tersebut ( $\sum_{i \in I} (R_{u,i} * S_{i,j})$ ). Selanjutnya menghitung total keseluruhan dari nilai *similarity* *item* yang mirip dengan *item* yang akan prediksi ( $\sum_{i \in I} |S_{i,j}|$ ). Langkah terakhir yaitu membagi langkah pertama dengan langkah kedua, yang mana akan menghasilkan nilai prediksi untuk *user* terhadap *item<sub>j</sub>* ( $P(u, j)$ ).

## 2.2. Skala Penilaian

Skala penilaian sering digunakan peneliti untuk mengetahui pendapat responden terhadap bahan penelitian yang dilakukan. Peneliti meminta responden untuk memberikan penilaian terhadap sekumpulan *item*. Hal ini dilakukan untuk mengetahui seberapa besar kesukaan responden terhadap *item* tersebut.

Ada beberapa jenis skala penilaian yang dapat digunakan pada saat penelitian, yaitu: skala *Likert*, skala *Guttman* dan skala *Rating*

### **2.2.1. Skala *Likert***

Skala *Likert* adalah skala yang dapat dipergunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang mengenai suatu gejala atau fenomena pendidikan. Dalam skala *Likert* terdapat dua bentuk pernyataan yaitu pernyataan positif yang berfungsi untuk mengukur sikap positif, dan pernyataan *negative* yang berfungsi untuk mengukur sikap *negative* objek sikap (Babbie, 2012).

Skala yang digunakan pada skala *likert* yaitu antara 1 sampai 5 dimana tiap nilai diartikan sebagai berikut:

1. Sangat tidak setuju
2. Tidak setuju
3. Netral
4. Setuju
5. Sangat setuju

### **2.2.2. Skala Guttman**

Skala Guttman merupakan skala yang menginginkan tipe jawaban tegas, seperti jawaban benar-salah, ya-tidak, pernah-tidak pernah, positif-*negative*, tinggi-rendah, baik-buruk, dan seterusnya. Pada skala Guttman, hanya ada dua interval, yaitu setuju dan tidak setuju (Babbie, 2012).

Skala Guttman dapat dibuat dalam bentuk pilihan ganda maupun daftar checklist. Untuk jawaban positif seperti benar, ya, tinggi, baik, dan sebagainya diberi skor 1; sedangkan untuk jawaban *negative* seperti salah, tidak, rendah, buruk, dan sebagainya diberi skor 0.

### **2.2.3. *Rating Scale***

*Rating scale* merupakan skala penilaian yang lebih fleksibel, skala penilaian ini tidak hanya untuk mengukur sikap tetapi dapat juga digunakan untuk mengukur persepsi responden terhadap fenomena lingkungan, seperti skala untuk mengukur status sosial, ekonomi, pengetahuan, kemampuan, dan lain-lain.

Pada skala penilaian *rating scale* menurut (Wimmer dkk,2006) keputusan peneliti yang menentukan skala mana yang digunakan untuk penilaian yang digunakan, apakah 1 sampai 3, 1 sampai 5, 1 sampai 7, 1 sampai 10 atau 1 sampai 100. Memilih jenis skala *rating* sebagian besar merupakan masalah preferensi pribadi. Namun ada beberapa hal yang harus dipertimbangkan, yaitu:

1. Skala dengan nilai yang lebih tinggi akan menggambarkan perbedaan penilaian *rating* yang lebih detail pada *item* jika dibandingkan dengan skala dengan penilaian rendah. Sebagai contoh ketika kita ingin menilai pentingnya suatu program dalam acara *weekday radio show*, jika responden mengatakan “semakin tinggi nilai maka semakin penting program tersebut untuk mereka”. Apakah skala 1-3 atau 1-10 yang digunakan. Tentu saja 1-10 karena skala ini menggambarkan perbedaan paling lebar (Wimmer dkk,2006).
2. Penelitian yang dilakukan (Wimmer dkk,2006) menyatakan bahwa laki-laki, perempuan, di semua tingkatan usia, semua suku dan bangsa menyukai penilaian 1-10. Hal ini dikarenakan skala 1-10 digunakan secara universal, terutama pada acara olahraga seperti pertandingan olimpiade. Hampir setiap orang mengerti skala 1-10 dimana 10 merupakan penilaian sempurna atau yang terbaik sedangkan 1 merupakan penilaian yang paling buruk. Penelitian (Wimmer dkk,2006) juga menunjukkan bahwa peneliti seharusnya tidak menggunakan skala *rating* 0-9 atau 1-9 karena responden umumnya tidak mengenali kalau 9 merupakan nilai tertinggi.

### **2.3. Evaluasi Sistem Rekomendasi**

Evaluasi digunakan untuk mengetahui seberapa bagus hasil rekomendasi yang diberikan sistem kepada *user*. Beberapa cara yang dapat dilakukan untuk mengevaluasi sistem rekomendasi yaitu dengan menghitung tingkat *error* pada hasil rekomendasi dan juga menghitung nilai presisi dari hasil UAT (*User Acceptance Test*) yang diberikan *user*.

### 2.3.1. Jenis-Jenis Perhitungan *Error*

Yang perlu diperhatikan dalam sistem rekomendasi adalah seberapa efektifkah hasil rekomendasi yang diberikan. Untuk mengukur tingkat akurasi hasil rekomendasi dapat dilakukan dengan melihat nilai error pada hasil rekomendasi yang diberikan. Terdapat banyak metode perhitungan *error* seperti :MSE (*Mean Square Error*), MAPE (*Mean Absolute Percentage Error*) dan MAE (*Mean Absolute Error*).

#### a. MSE (*Mean Square Error*)

MSE adalah rata-rata absolut dari kesalahan peramalan yang dikuadratkan. Rumus dari MSE adalah sebagai berikut: (Soedjianto, 2006).

$$MSE = \frac{\sum_{t=1}^m |f_t - \hat{f}_t|^2}{n} \quad \dots(2.6)$$

keterangan:

- $f_t$  = permintaan aktual periode t
- $\hat{f}$  = ramalan permintaan periode t
- $m$  = jumlah periode peramalan

MSE biasa dipakai untuk menghitung hasil dari peramalan namun MSE juga dapat diterapkan pada perhitungan hasil rekomendasi yaitu dengan mengganti  $f_t$  (permintaan aktual periode t) menjadi *rating* user terhadap item  $\hat{f}_t$  (ramalan permintaan periode t) menjadi prediksi *rating* user terhadap item t dan m menjadi jumlah item yang dihitung.

#### b. MAPE (*Mean Absolute Percentage*)

MAPE adalah rata-rata persentase absolut dari kesalahan peramalan dengan menghitung *error* absolut tiap periode. *Error* ini kemudian dibagi dengan n. Rumus dari MAPE ini adalah sebagai berikut: (Soedjianto, 2006)

$$MAPE = \frac{\sum_{t=1}^m \left( \frac{|f_t - \hat{f}_t|}{f_t} \right) \cdot 100\%}{m} \quad \dots(2.7)$$

c. MAE (*Mean Absolute Error*)

Dari ketiga jenis perhitungan *error*, MAE merupakan persamaan perhitungan nilai *error* yang paling sering digunakan untuk menghitung nilai *error* dari hasil rekomendasi. *Mean absolute error* (MAE) merupakan persamaan yang termasuk jenis *statistical accuracy metrics* dimana MAE akan menghitung nilai rata-rata selisih antara nilai prediksi dengan nilai yang sebenarnya.

Persamaan MAE (Sarwar, 2001):

$$MAE = \frac{\sum_{i=1}^n |p_i - q_i|}{N} \quad \dots(2.8)$$

MAE = nilai rata-rata kesalahan hitungan

N = jumlah *item* yang dihitung

$p_i$  = nilai prediksi *item* ke  $i$

$q_i$  = nilai *rating* sebenarnya *item* ke  $i$

Semakin rendah nilai MAE yang didapat, maka sistem semakin akurat dalam memprediksi *rating* untuk setiap *user*.

### 2.3.2. Presisi (*Precision*)

Sistem rekomendasi memberikan sekumpulan *item* sebagai hasil rekomendasi untuk *user*. Dari beberapa *item-item* tersebut tentu tidak semua *item* yang relevan atau yang sesuai dengan kebutuhan *user*. Untuk mengetahui kualitas hasil rekomendasi, dapat menggunakan rumus relevansi presisi yang membandingkan antara *item* yang relevan dengan total *item* yang dihasilkan atau yang direkomendasikan kepada *user*.

Persamaan *precision* (Laoli, 2009):

$$Precision = \frac{\# \text{relevant items retrieved}}{\# \text{(retrieved items)}} = P(\text{relevant} | \text{retrieved}) \dots(2.9)$$



*Relevant items retrieved* adalah jumlah *item* relevan yang direkomendasikan sedangkan *retrieved items* adalah jumlah total *item* yang direkomendasikan.

## 2.4. *Sparsity*

*Sparsity* merupakan kondisi dimana terdapat banyak kekosongan *rating* pada *item* sehingga nilai *rating* *user* terhadap *item* menjadi jarang/renggang. Hal ini disebabkan karena *user* cenderung memberi penilaian terhadap beberapa buku saja. Masalah *sparsity* merupakan kendala yang ada pada sistem rekomendasi dengan metode *collaborative filtering* karena pada metode *collaborative filtering* *rating* merupakan hal yang paling penting dalam pemberian rekomendasi. Semakin banyak *rating* yang terkumpul maka semakin baik hasil rekomendasi yang diberikan. Semakin tinggi *sparsity* data *rating* maka akan memberikan *poor accuracy* (akurasi yang rendah) terhadap hasil rekomendasi (Xue, 2009) pada (Sanjung, 2011).

Jumlah *user* dan jumlah *item* yang besar akan sangat membantu proses perekomendasi, namun tidak mungkin semua *user* akan aktif menggunakan sistem dan memberikan penilaian pada setiap *item*. Terkadang *user* merasa malas untuk member nilai *rating* pada *item* setelah memberi *rating* terhadap beberapa *item* saja. Pada umumnya *user* hanya melakukan *rating* terhadap *item-item* dalam jumlah yang lebih kecil dari jumlah *item* yang tersedia di *database*, sehingga menyebabkan matriks *user-item* mengalami kekosongan data yang sangat tinggi (Budianto, 2012).

Banyak cara yang dilakukan untuk mengatasi masalah *sparsity* ini salah satunya yaitu dengan mengumpulkan data penilaian *rating* *user* secara implisit. *User* tidak perlu memberikan penilaian secara langsung dengan memilih *rating* penilaian yang disediakan. *User* hanya cukup melakukan aktifitas terhadap sistem seperti melakukan klik, *copy*, *drag*, *save* dan lain-lain dimana aktifitas-aktifitas ini dicatat sistem kemudian dikonversikan menjadi data *rating*. Selain itu ada juga teknik *smoothing*, teknik ini mengizinkan sistem untuk mengisi *rating-rating* yang masih kosong sehingga kekosongan *rating* tersebut dapat teratasi.

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1. Data Penelitian**

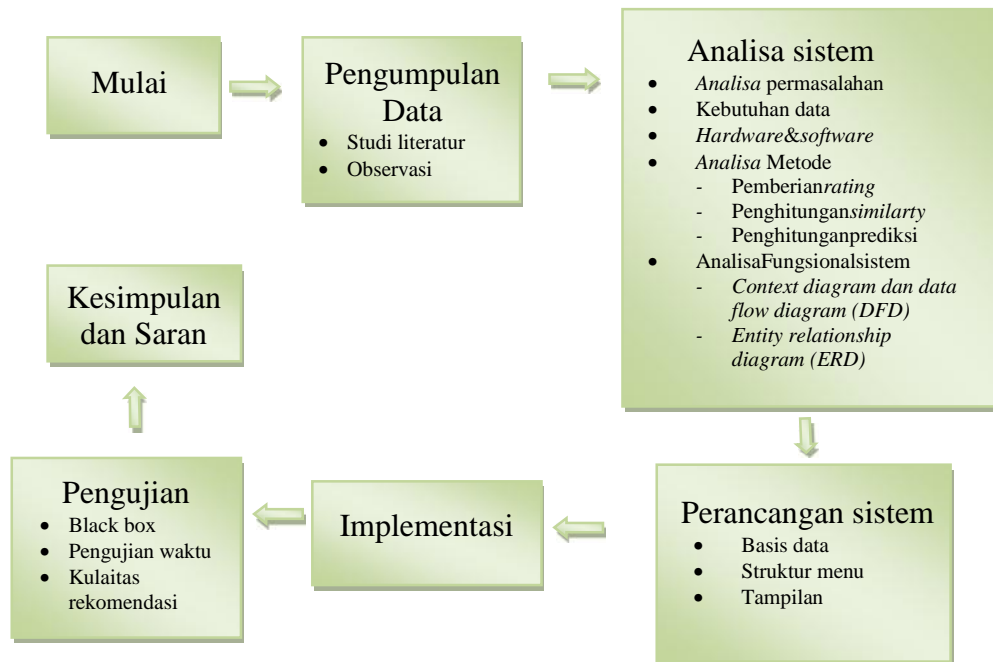
Data yang digunakan pada penelitian ini merupakan data yang terdapat di perpustakaan Fakultas Sains dan Teknologi. Data yang digunakan adalah data buku, mahasiswa (angkatan 2008-2011), peminjaman dan pencarian. Data-data buku, mahasiswa dan peminjaman diperoleh dengan meng-*export database* sistem yang terdapat di perpustakaan Fakultas Sains dan Teknologi, sedangkan untuk data pencarian diperoleh dengan cara meminta beberapa mahasiswa untuk melakukan pencarian buku pada sistem yang telah dibangun.

Jumlah data yang terhimpun yaitu sebanyak 2010 data buku, data mahasiswa yang terhimpun sebanyak 789 data, peminjaman sebanyak 795 data dan pencarian sebanyak 271 data. Dari data peminjaman dan pencarian diperoleh data *rating* sebanyak 841 data.

#### **3.2. Tahapan Penelitian**

Tahapan penelitian yang dilakukan dalam merancang bangun sebuah sistem rekomendasi buku perpustakaan dengan metode *collaborative filtering* dapat dilihat pada Gambar 3.1.

Penelitian dimulai dengan mengumpulkan data melalui studi literature dan melakukan observasi ke perpustakaan Fakultas Sains dan Teknologi. Tahap selanjutnya yaitu melakukan analisa kemudian dilakukan perancangan dan implementasi. Tahap terakhir yaitu pengujian untuk menguji sistem yang dibangun.



Gambar 3.1 Tahapan Penelitian

### 3.2.1. Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data-data yang diperlukan untuk mendukung penelitian. aktivitas yang dilakukan pada tahapan pengumpulan data yaitu

- Studi literatur, pada tahapan ini mengumpulkan data-data berupa teori mengenai perpustakaan, sistem rekomendasi, tahapan dalam membangun sistem rekoendasi, metode *collaboraive filtering* yang digunakan dalam pengembangan sistem dan algoritma-algoritma yang digunakan pada sistem.
- Observasi, pada tahapan ini penulis melakukan observasi ke perpustakaan Fakultas Sains dan Teknologi UIN Suska Riau melihat data buku serta laporan tugas akhir dan kerja praktek.

### 3.2.2. Analisa

Tahap ini mengidentifikasi seluruh masalah yang terjadi kemudian melakukan perancangan sesuai dengan analisa yang telah dilakukan sebelumnya.

Analisa yang dilakukan yaitu:

a. Analisa Permasalahan.

Pada tahap ini akan dijelaskan mengenai permasalahan yang terjadi kemudian bagaimana tahapan-tahapan penyelesaian masalah tersebut.

b. Analisa kebutuhan data.

Pada tahapan ini akan ditemukan kebutuhan data pada sistem rekomendasi yang akan dibuat.

c. Analisa kebutuhan perangkat keras dan perangkat lunak.

Tahapan ini membahas tentang perangkat keras dan perangkat lunak yang dibutuhkan untuk membangun dan mengimplementasikan sistem rekomendasi buku perpustakaan ini.

d. Analisa Metode

i. Pemberian *rating*, tahap ini akan dibahas mengenai dasar pemberian *rating*, bagaimana *user* member *rating* dan kemudian dianalisa skala pemberian *rating* yang digunakan pada sistem.

ii. Penghitungan kemiripan/*Similarity*, pada tahap ini dibahas metode *adjusted cosine similarty* untuk menemukan *item* yang mirip dengan *item* yang pernah dipinjam/dicari oleh *user*.

iii. Penghitungan prediksi, setelah ditemukan *item* yang mirip lalu setiap *item* diprediksi nilai *rating*-nya. Buku yang memiliki nilai prediksi tertinggi akan direkomendasikan kepada *user*.

e. Analisa fungsional sistem,

i. *Context diagram* dan *data flow diagram* (DFD) yang digunakan untuk menggambarkan aliran data dari sistem rekomendasi buku perpustakaan,

ii. *Entity relationship diagram* (ERD) yang digunakan untuk menggambarkan hubungan antara data satu dengan data lainnya.

f. Analisa struktur menu dan alur sistem.

### 3.2.3. Perancangan

Setelah selesai melakukan tahap analisa selajutnya dilakukan perancangan berdasarkan hasil analisa tersebut. Perancangan ini mencakup perancangan basis data, struktur menudan tampilan. Pada perancangan basisdata disesuaikan data yang terdapat pada perpustakaan fakultas sains dan teknologi dengan penambahan data daftar isi buku yang berguna untuk pencarian buku. Perancangan struktur menu memberikan gambaran menu yang dapat diakses *user*. Perancangan tampilan berisi rancangan tampilan untuk *admin* dan untuk *user*

### 3.2.4. Implementasi

Tahap implementasi akan dilakukan setelah tahap analisa dan perancangan selesai. Pada sistem rekomendasi buku perpustakaan ini akan menggunakan pemrograman PHP versi 5 karena PHP dapat membangun sebuah sistem berbasis web yang ringan dan handal. Sistem dirancang berbasis web agar dapat dijalankan secara *client-server* sehingga ketika ada perubahan data cukup dilakukan disisi *server* dan hasilnya dapat dilihat disisi *client*.

Implementasi pengembangan aplikasi ini akan dikembangkan pada spesifikasi *hardware* dan *software* berikut:

1. Perangkat keras

Processor : *Intel(R) core(TM) i5-2430M CPU @ 2.40 GHz*  
Memori (RAM) : 2.00 GB

2. Perangkat Lunak

Sistem operasi : *Windows 7 Ultimate 64-bit*  
Bahasa pemrograman : *PHP Script Language Version 5.2.6.*  
*Tools* perancangan : *Notepad ++ versi 5.9.3.*  
*Tools*DBMS : *phpMyAdmin Database Manager Version 2.10.3.*  
*web server* : *Apache Web Server Version 2.2.8.*  
*database* : *MySQL Database Version 5.0.51b.*

### 3.2.5. Pengujian

Sementara untuk tahapan pengujian yang akan dilakukan pada rekomendasi buku perpustakaan yang telah dibangun meliputi :

1. Pengujian *blackbox* untuk pengujian tingkah laku aplikasi yang telah dirancang. Pengujian ini bertujuan untuk memastikan bahwa sistem yang dibangun telah berjalan sesuai dengan analisa dan perancangan.
2. Pengujian terhadap kualitas rekomendasi ini bertujuan untuk menilai seberapa baik kualitas rekomendasi buku yang dihasilkan. Pengujian ini dilakukan dengan menggunakan UAT (*User Acceptance Test*) dan fungsi perhitungan *error*.
  - i. Pada UAT, sistem dinilai langsung oleh petugas perpustakaan dan mahasiswa dimana mereka diberi kuisisioner yang didalamnya berisi beberapa pertanyaan seputar sistem rekomendasi yang telah dibangun.
  - ii. Fungsi perhitungan *error* yang digunakan adalah *mean absolute error* (MAE). Perhitungan ini tidak terlalu rumit dan tidak memberatkan sistem karena hanya mencari selisih antara *rating* asli dengan *rating* prediksi kemudian membaginya dengan jumlah *item* yang dihitung. Selain itu, berdasarkan sumber-sumber yang ada rata-rata untuk pengujian sistem rekomendasi dengan metode *collaborative filtering* menggunakan MAE.
- c. Pengujian waktu eksekusi sistem, pengujian ini bertujuan untuk melihat seberapa besar waktu yang dibutuhkan untuk menghasilkan rekomendasi buku. Sistem akan diuji dengan memasukkan data buku-buku dalam jumlah tertentu kemudian dihitung waktu yang diperlukan untuk memperoleh hasil rekomendasi. Selain itu, waktu eksekusi sistem juga diuji berdasarkan tingkat *sparsity* dari *dataset* yang digunakan.

### 3.2.6. Kesimpulan dan Saran

Pada bagian ini, berisi kesimpulan mengenai hasil evaluasi dari seluruh kegiatan yang dilakukan dalam melakukan penelitian terhadap Analisa dan Penerapan metode *collaborative filtering* dalam sistem rekomendasi buku

perpustakaan. Selain itu juga dijelaskan mengenai hal-hal yang mempengaruhi proses perekomendasi buku baik dari segi kualitas hasil rekomendasi maupun dari segi waktu eksekusi yang dibutuhkan untuk memproses sistem. Pada tahap ini juga diberikan saran-saran untuk pengembangan dan pengelolaan sistem lebih lanjut.

## **BAB IV**

### **ANALISA DAN PERANCANGAN**

#### **4.1. Analisa Sistem Rekomendasi**

Dalam analisa sistem rekomendasi buku perpustakaan ini akan dibahas mengenai analisa permasalahan, analisa kebutuhan data, analisa perangkat lunak dan perangkat keras, analisa metode, dan analisa fungsional .

##### **4.1.1. Analisa Permasalahan**

Berdasarkan penjabaran yang telah dipaparkan pada bab pendahuluan, dapat diketahui bahwa permasalahan yang sering terjadi di perpustakaan yaitu dikarenakan banyaknya jumlah buku yang terdapat di perpustakaan. Jumlah buku yang banyak tersebut membuat mahasiswa sering kesulitan untuk menemukan buku yang tepat dan sesuai dengan kebutuhan mereka.

Salah satu solusi yang dapat dilakukan untuk mengatasi masalah tersebut yaitu membangun sebuah sistem rekomendasi buku. Sistem dapat menganalisa peminjaman dan pencarian yang pernah dilakukan *user active* (mahasiswa) kemudian membandingkannya dengan catatan peminjaman dan pencarian yang pernah dilakukan oleh orang lain. Dari hasil tersebut didapatkan korelasi antar buku. Sistem akan membandingkan buku yang pernah dipinjam (buku lama bagi *user active*) dan yang belum pernah dipinjam/dicari (buku baru bagi *user active*) oleh *user active*. Buku-buku baru yang belum pernah dipinjam yang saling berkorelasi dengan buku lama akan dilakukan perhitungan kemiripan dan nilai prediksi *rating* sehingga menghasilkan himpunan buku baru yang dapat direkomendasikan kepada mahasiswa.

Sistem rekomendasi buku perpustakaan ini nantinya akan dibagi menjadi 2 tingkatan hak akses yaitu: *administrator* dan mahasiswa. *Administrator* bertugas



untuk menginputkan data buku, pencarian dan peminjaman, *administrator* juga bertugas mengkonversikan data peminjaman dan data pencarian menjadi data *rating*. Konversi data dilakukan *administrator* hanya untuk pertama kali ketika data baru diimpor karena untuk selanjutnya sistem secara otomatis akan mengkonversi data pencarian dan peminjaman ketika mahasiswa melakukan pencarian dan peminjaman. Sedangkan hak akses mahasiswa merupakan *user* yang akan menggunakan sistem untuk melakukan pencarian dan peminjaman lalu mendapatkan hasil rekomendasi buku berdasarkan pencarian dan peminjaman yang pernah mereka lakukan.

Sistem rekomendasi buku dengan metode *collaborative filtering* yang akan dibangun secara umum memiliki 2 tahapan dalam pemberian rekomendasi buku kepada mahasiswa. Tahapan tersebut yaitu : penentuan kemiripan (*similarity*) antar *item*, setelah mendapatkan *item* yang mirip kemudian dihitung prediksi *rating user* terhadap *item*/buku tersebut. Buku yang memiliki nilai prediksi yang tinggi akan direkomendasikan kepada mahasiswa. Untuk lebih jelasnya Gambar 4.1 menggambarkan sistem secara keseluruhan.



Gambar 4.1. Deskripsi Umum Sistem

Pada mulanya *administrator* meng-input-kan data seperti data buku dan peminjaman yang masih berupa *file excel* kedalam sistem. Sistem juga menerima masukan (*input*) dari *user*(mahasiswa) berupa data pencarian dan peminjaman. Data

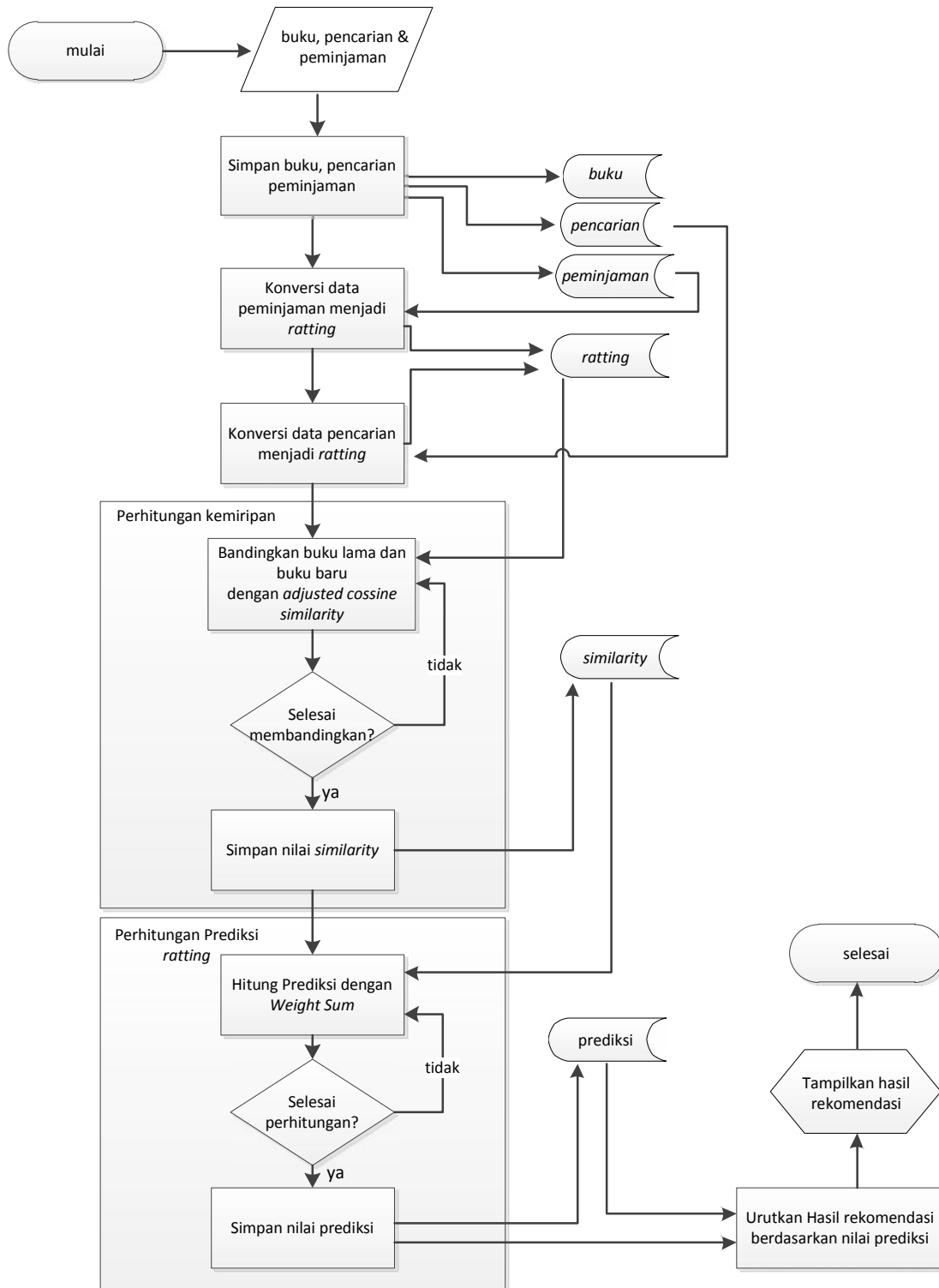
ini akan dikonversi menjadi nilai *rating* terhadap suatu buku. Nilai *rating* diperoleh berdasarkan aktivitas *user* terhadap buku.

Setelah diperoleh nilai *rating*, dihitung kemiripan antar buku dengan algoritma *adjusted cosine similarity*. Setelah nilai kemiripan antar buku diperoleh, proses selanjutnya adalah menghitung prediksi *rating user* terhadap buku-buku yang mirip dengan algoritma *weight sum*. Setelah didapat nilai prediksi, maka buku-buku yang memiliki nilai prediksi tertinggi yang akan direkomendasikan kepada *user*.

Agar lebih jelas mengenai alur sistem yang akan dibangun dapat dilihat pada Gambar 4.2 yang menjelaskan tentang tahapan-tahapan dalam pemberian rekomendasi buku dengan menggunakan metode *collaborative filtering*. Tahapan dimulai dengan mengubah data peminjaman dan pencarian menjadi nilai *rating* agar dapat diolah. Data dikonversi dengan aturan pengkonversian yang telah dijabarkan sebelumnya. Tahapan ini akan menghasilkan data *rating* yang kemudian disimpan kedalam *database* untuk proses selanjutnya.

Data *rating* buku baru (buku yang belum pernah di-*rating*) ini kemudian dibandingkan satu persatu dengan data *rating* buku lama (buku yang pernah di-*rating*) dengan menggunakan persamaan (2.3). Hasil akhir tahapan ini adalah nilai kemiripan antar buku baru dan buku lama yang berkorelasi. Nilai kemiripan (*similarity*) kemudian disimpan ditabel *similarity*. Tabel ini bersifat *temporary* karena digunakan hanya untuk menampung sementara nilai *similarity*. Ketika ada proses perhitungan nilai kemiripan yang lain tabel ini akan dikosongkan.

Dari nilai *similarity* yang diperoleh kita dapat menentukan buku mana yang mempunyai kemiripan yang sangat kuat yaitu buku yang memiliki nilai *similarity* paling tinggi. Untuk tahap selanjutnya, data dipanggil dari tabel *similarity* diurut secara *descending* kemudian dihitung nilai prediksi *rating* dengan menggunakan persamaan (2.5). Hasil prediksi *rating* disimpan di tabel prediksi yang bersifat *temporary* juga.



Gambar 4.2 Flowchart Sistem Rekomendasi Buku Perpustakaan

Setelah seluruh perhitungan prediksi selesai, buku diurutkan berdasarkan nilai prediksi. Buku yang memiliki nilai prediksi tertinggi akan diletakkan di urutan pertama. Hasil rekomendasi buku ditampilkan berdasarkan nilai prediksi yang telah diurutkan.

#### **4.1.2. Analisa Kebutuhan Data**

Pada pembuatan aplikasi sistem rekomendasi ini data yang dibutuhkan dapat dijabarkan secara umum sebagai berikut:

a. Administrator

Administrator berisi data *administrator* yang berperan sebagai pengelola sistem. Data ini digunakan untuk proses verifikasi agar dapat mengakses sistem. Data ini mencakup *username*, dan *password*, nama lengkap dan nomor telepon.

b. Mahasiswa

Selain *administrator* juga disediakan pengguna mahasiswa yang menggunakan sistem untuk memperoleh rekomendasi buku. Data ini mencakup nomor mahasiswa/NIM (Nomor Induk Mahasiswa) untuk *login* dan data diri mahasiswa seperti nama, status, alamat, kelamin, tempat lahir dan tanggal lahir.

c. Buku

Buku berisi data buku-buku yang akan direkomendasikan. Data ini mencakup id buku, judul buku, jumlah *exemplar*, tanggal *input*, ISBN, penerjemah, editor, penulis, cetakan, penerbit, tahun terbit, kota, jumlah halaman, panjang lebar, subyek, daftar isi/pembahasan dan catatan umum.

d. Pencarian

Pencarian berisi catatan buku-buku yang pernah dicari/diklik oleh *user* ketika berinteraksi dengan sistem beserta jumlah klik yang telah dilakukannya. Data yang mencakup id pencarian, nomor mahasiswa, id buku, dan jumlah klik ini akan dijadikan nilai *rating user* terhadap suatu buku.

e. Peminjaman

Peminjaman berisi catatan peminjaman *user* terhadap suatu buku. Data yang mencakup id pinjam, nomor *barcode*, judul, tanggal pinjam, nomor mahasiswa dan tanggal kembali ini akan digunakan sebagai *rating user*.

f. *Rating*

*Rating* berisi nilai *rate* yang diberi *user* terhadap buku. Nilai ini diperoleh dari aktifitas *user* baik berupa pencarian/klik dan juga peminjaman *user*.

g. *Similarity*

*Similarity* berisi data kemiripan yang diperoleh dari perhitungan rumus *adjusted cosine*. Data ini dipergunakan untuk tahap penentuan prediksi dimana data tersebut diurutkan dari yang besar ke yang kecil, kemudian diambil sebanyak *n* data untuk dilakukan tahap perhitungan prediksi.

h. Prediksi

Prediksi berisi data buku-buku yang belum pernah di-*rating user* beserta nilai prediksi *rating*-nya. Data ini kemudian diurutkan dari yang besar ke yang kecil. Kemudian sebanyak *n* data diambil untuk direkomendasikan ke *user*.

#### 4.1.3. Analisa Kebutuhan Perangkat Keras dan Perangkat Lunak

Agar dapat bekerja dengan baik, sistem membutuhkan dukungan perangkat keras dan perangkat lunak untuk menjalankan sistem tersebut. Berikut analisa perangkat keras dan perangkat lunak yang dibutuhkan untuk membangun dan mengimplementasikan sistem rekomendasi buku.

1. Perangkat keras

Processor : *Intel Pentium IV, 2.5 GHz*

Memori : 1 GB

2. Perangkat Lunak

Sistem operasi : *Ms Windows XP/ Windows 7*

Basis Data : [\*MySQL Database Version 5.0.51b\*](#).

<i>Program tool</i>	: <i>Notepad ++ v 5.8.2</i>
<i>Web Server</i>	: <a href="#"><i>Apache Web Server Version 2.2.8</i></a>
<i>Web Browser</i>	: <i>Google Chrome version 19.0.1084.52</i>

#### **4.1.4. Analisa Metode**

Seperti yang telah dijelaskan pada bab-bab sebelumnya mengenai *Collaborative Filtering*, pada subbab ini akan dijabarkan mengenai tahapan kerja dari aplikasi sistem rekomendasi buku perpustakaan yang akan dibangun sesuai dengan metode *collaborative filtering*. Menurut (Sarwar,2001) tahapan metode *collaborative filtering* terbagi menjadi 2 tahapan yaitu : perhitungan kemiripan dan perhitungan prediksi.

##### **4.1.4.1. Pemberian Rating**

*Rating* yang digunakan pada sistem rekomendasi buku perpustakaan diperoleh secara implisit dimana *user* memberi *rating* secara tidak sadar. Setiap pencarian dan peminjaman yang dilakukan *user* akan dihitung kemudian dijadikan sebagai *rating* dengan catatan bahwa *rating* yang diperoleh dari peminjaman nilainya lebih tinggi dari pencarian karena ketika *user* meminjam buku, berarti *user* telah melihat dan membaca isi buku. Berbeda dengan pencarian, dimana *user* hanya sebatas melihat penjelasan singkat mengenai buku tersebut jika *user* merasa buku tersebut cocok maka ia akan melakukan peminjaman pada buku tersebut.

Untuk skala pemberian *rating* awalnya skala yang digunakan berdasarkan skala Likert yaitu 1-5 dimana nilai *rating* dipecah menjadi 1-3 adalah *rating* untuk pencarian dan 4-5 adalah *rating* peminjaman. Namun pemberian skala *rating* seperti ini terdapat kekurangan karena *rating* untuk peminjaman terlalu sempit sehingga pada proses penghitungan nilai kemiripan banyak menghasilkan nilai 0. Hal ini akan mempengaruhi hasil penghitungan prediksi. Semakin banyak nilai kemiripan yang 0 maka semakin sedikit data pembandingan yang digunakan untuk perhitungan prediksi karena nilai kemiripan 0 tidak diikuti dalam perhitungan prediksi.

Karena masalah tersebut dilakukan pelebaran *rating* berdasarkan *rating scale* menurut (Wimmer,2006)menjadi 1-10, dimana data *rating* dibagi menjadi 2 yaitu 1-5 adalah data pencarian dan 6-10 adalah data peminjaman. Hal ini dilakukan karena nilai *rating* yang diperoleh melalui pencarian tidak sebesar nilai *rating* yang diperoleh melalui peminjaman. Selain itu juga berdasarkan penelitian yang dilakukan (Roger dkk,2006) skala *rating* 1-10 (*one to ten*) merupakan skala penilaian yang umum diketahui masyarakat untuk nilai *rating* 10 menggambarkan penilaian yang terbaik sedangkan 1 merupakan penilaian yang terburuk.

Berdasarkan skala *rating* 1-10 dilakukan pembagian *rating* sebagai berikut:

1. Nilai *rating* 1 sampai 5 diberikan ketika *user* mencari dan melakukan klik pada buku. Jika buku tersebut diklik sebanyak 1 kali, maka nilai *rating user* terhadap bukutersebut adalah 1. Jika *user* melakukan klik sebanyak 2 kali, nilai *rating user* terhadap bukudi-*update* menjadi 2. Begitu seterusnya sampai nilai *rating* menjadi 5. Ketika *user* melakukan klik lebih dari 5 kali, nilai *rating* tetap 5 karena nilai *rating* maksimal yang diperoleh melalui klik/pencarian adalah 5.
2. Nilai *rating* 6 sampai 10 diberikan ketika *user* melakukan peminjaman pada *item*/buku. Dimana ketika *user* meminjam buku 1 kali, nilai *rating user* terhadap buku tersebut adalah 6. Ketika *user* meminjam kembali bukutersebut, berarti jumlah peminjaman *user* terhadap buku sebanyak 2 kali dan nilai *rating* di-*update* menjadi 7. Begitu seterusnya sampai nilai *rating* mencapai 10. Ketika *user* meminjam buku lebih dari 5 kali, nilai *rating* tetap 10 karena maksimal nilai *rating* yang diberi *user* adalah 10.

#### **4.1.4.2. Perhitungan Kemiripan (*Similarity*)**

Pada penelitian ini penulis menggunakan persamaan *adjusted cosine similarity*(2.3) untuk menghitung kemiripan karena merunut dari penelitian yang dilakukan (Sarwar,2001), algoritma yang dapat menghasilkan nilai *error*(MAE) terkecil adalah algoritma *adjusted cosine similarity*.

Untuk menghitung nilai kemiripan dibutuhkan data *rating user* dimana nilai *rating* diperoleh secara implisit, *user* tidak menyadari bahwa ia telah memberikan *rating* kepada sistem. Untuk *rating item* diasumsikan bahwa nilai 0 *user* belum pernah mencari atau meminjam buku tersebut. Nilai *rating* 1-5 jika *user* pernah melakukan pencarian dan mengklik 1 sampai 5 kali buku tersebut, nilai *rating* tetap 5 ketika *user* melakukan klik lebih dari 5 kali. *Rating* bernilai 6-10 jika *user* telah meminjam sebanyak 1 sampai 5 kali dan *rating* tetap bernilai 10 ketika *user* meminjam buku lebih dari 5 kali.

Langkah awal dalam proses rekomendasi yaitu menemukan buku yang telah di-*rating user*, hal ini dilakukan agar buku-buku tersebut tidak diikutkan dalam perhitungan rekomendasi. Buku yang direkomendasikan adalah buku yang belum pernah dipinjam atau dicari oleh *user*.

Setelah didapatkan *list* buku yang belum pernah di-*rating user*, untuk setiap buku tersebut dihitung nilai kemiripannya dengan buku yang pernah di-*rating user*. Misalnya terdapat 5 buah buku a,b,c,d,e. *User* telah me-*rating* buku a dan e, berarti buku b,c dan d belum di-*rating user*. Nilai kemiripan yang dihitung adalah antara buku b - a, b - e, c - a, c - e, d - a, dan d - e.

Sebagai contoh perhitungan proses pemberian rekomendasi pada sistem, diberikan 4 buah *sample* buku dan 4 *user* dimana pada tabel 4.1 dapat dilihat nilai *rating* yang diberikan *user* kepada masing-masing buku.

Tabel 4.1 Nilai *Rating user* Terhadap *Item*

Buku	A	B	C	D	$\bar{R}_{u..}$
Hardi	6		1	8	5
Riesti	2	10			6
Nina		3	1		2
Veriza	2	3	2	1	2

Pada tabel diatas A merupakan perwakilan buku Sistem Operasi, B adalah perwakilan buku ‘Algoritma Pemrograman C++’, C merupakan perwakilan dari buku



‘Just XML’, dan D merupakan perwakilan buku ‘Membuat Sendiri Aplikasi Android Untuk Pemula’. Diasumsikan *user* yang sedang *login* adalah Hardi. Untuk menghitung nilai kemiripan menggunakan persamaan *adjusted cosine similarity* (2.3), terlebih dahulu kita tentukan pasangan *item* yang akan dihitung nilai kemiripannya. Nilai kemiripan yang dihitung adalah antara *item* yang sudah pernah di-*rating user* dan *item* yang belum pernah di-*rating*. Untuk kasus *user* Hardi pada tabel 4.1, *item* yang akan dihitung nilai kemiripannya yaitu A-B, C-B, dan D-B.

Nilai kemiripan antara buku A (Sistem Operasi) dan B (Algoritma Pemrograman C++) dapat dilihat pada perhitungan dibawah ini.

$$\begin{aligned} \text{Sim(A,B)} &= \frac{\sum_{u \in U} (R_{u,a} - \bar{R}_u)(R_{u,b} - \bar{R}_u)}{\sqrt{\sum_{u \in U} (R_{u,a} - \bar{R}_u)^2} \sqrt{\sum_{u \in U} (R_{u,b} - \bar{R}_u)^2}} \\ &= \frac{(2-6)(10-6) + (2-2)(3-2)}{\sqrt{(2-6)^2 + (2-2)^2} \sqrt{(10-6)^2 + (3-2)^2}} \\ &= \frac{(-4)*4 + 0*(1)}{4*4.123} \\ &= \frac{-16}{16.49} = -0.97 \end{aligned}$$

Hasil dari perhitungan *similarity*/kemiripan disimpan didalam tabel *similarity* yang berguna nantinya untuk proses perhitungan prediksi.

Perhitungan nilai kemiripan antara buku C dan B dapat dilihat dibawah

$$\begin{aligned} \text{Sim(C,B)} &= \frac{\sum_{u \in U} (R_{u,a} - \bar{R}_u)(R_{u,c} - \bar{R}_u)}{\sqrt{\sum_{u \in U} (R_{u,a} - \bar{R}_u)^2} \sqrt{\sum_{u \in U} (R_{u,c} - \bar{R}_u)^2}} \\ &= \frac{(3-2)(1-2) + (3-2)(2-2)}{\sqrt{(3-2)^2 + (3-2)^2} \sqrt{(1-2)^2 + (2-2)^2}} \\ &= \frac{-1}{1.41*1} = -0.707 \end{aligned}$$

Perhitungan nilai kemiripan dilakukan jika terdapat 2 atau lebih *rating* dari *user* lain terhadap kedua *item* tersebut. Seperti perhitungan *sim(C,B)* diatas, *user* yang melakukan *rating* adalah Nina dan Veriza. Jika *rating user* lain terhadap buku

kurang dari 2 orang maka perhitungan nilai kemiripan tidak dapat dilakukan seperti perhitungan nilai kemiripan antara buku D dan B tidak dapat dihitung karena *user* yang me-*rating* kedua *item* tersebut hanyalah Veriza.

Dari perhitungan *similarity* diatas, diperoleh tabel kemiripan (*similarity*) yang dapat dilihat pada Tabel 2.2 dibawah ini,

Tabel 4.2 Nilai Kemiripan Antar *Item*

Sim(A,B)	-0.97
Sim(C,B)	-0.707

Tabel *similarity* diatas merupakan tabel *temporary* dimana tabel tersebut bersifat dinamis, data pada tabel ini akan berubah bergantung dari *user* yang *login*. Seperti Tabel 4.3 dibawah ini merupakan tabel *similarity* ketika *user* yang *login* adalah Riesti.

Tabel 4.3 Nilai Kemiripan Untuk *User* Riesti

Sim(A,C)	-1
Sim(A,D)	0.95
Sim(B,C)	-0.707

#### 4.1.4.3. Perhitungan Prediksi

Setelah didapatkan nilai kemiripan langkah selanjutnya adalah menghitung prediksi. Pertama sekali yang dilakukan dalam menghitung prediksi adalah membaca tabel *similarity* pada *database* dimana data diurutkan secara *descending* dari data yang besar ke yang kecil. Guna pengurutan/perangkingan ini adalah untuk mengambil sebanyak *n* data yang nilai *similarity*-nya tertinggi. Setelah itu hitung prediksi untuk masing-masing buku yang belum pernah di-*rating user* tersebut dengan persamaan *weight sum*(2,5). Contoh dibawah ini menunjukkan prediksi *rating* yang diberi *user* Hardi terhadap *item* B.

$$\begin{aligned}
P(\text{Hardi}, B) &= \frac{\sum_{i \in I} (R_{u,i} * S_{i,j})}{\sum_{i \in I} |S_{i,j}|} \\
&= \frac{((1 * -0.707) + (6 * -0.97))}{|-0.707| + |-0.97|} \\
&= \frac{-6.527}{1.677} = -3.89
\end{aligned}$$

Untuk setiap *item* yang mirip dengan *item* B dilakukan perhitungan diatas yang menghasilkan nilai prediksi *rating*. Nilai ini disimpan ke *database* dan nantinya akan ditampilkan sebagai *output*/hasil akhir dari sistem rekomendasi.

Berikut contoh lain perhitungan prediksi untuk *user* Riesti.

$$\begin{aligned}
P(\text{Riesti}, C) &= \frac{\sum_{i \in I} (R_{u,i} * S_{i,j})}{\sum_{i \in I} |S_{i,j}|} \\
&= \frac{((2 * -1) + (10 * -0.707))}{|-1| + |-0.707|} \\
&= \frac{-2}{1.707} = -5.31
\end{aligned}$$

$$\begin{aligned}
P(\text{Riesti}, D) &= \frac{\sum_{i \in I} (R_{u,i} * S_{i,j})}{\sum_{i \in I} |S_{i,j}|} \\
&= \frac{(2 * -0.95)}{|0.95|} \\
&= \frac{1.9}{0.95} = 2
\end{aligned}$$

Dari perhitungan diatas dapat kita lihat bahwa prediksi *rating user* Riesti untuk buku C adalah -5.31 dan prediksi *rating* Riesti terhadap buku D adalah 2. Data prediksi ini akan disimpan dalam sebuah tabel *temporary* yang bernama tabel prediksi. Setelah semua data prediksi *rating* terkumpul, dilakukan pengurutan berdasarkan nilai prediksi. Nilai prediksi tertinggi akan berada di urutan pertama, pada kasus prediksi *rating* Riesti terhadap buku D berada di urutan pertama kemudian urutan kedua buku C. Berarti hasil rekomendasi buku yang diberikan untuk *user* Riesti yaitu:

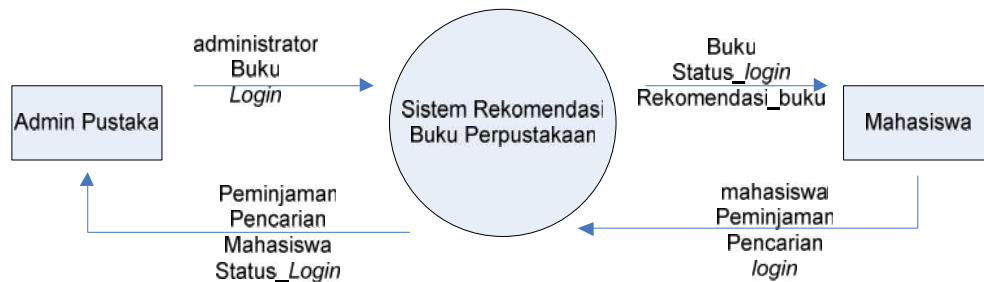
1. Membuat Sendiri Aplikasi Android Untuk Pemula (D)
2. Just XML(C)

#### 4.1.5. Analisa Fungsional Sistem

Pada analisa fungsional akan dijelaskan mengenai *context diagram* yang dapat dilihat pada Gambar 4.2 dan keterangannya dapat dilihat pada Tabel 4.7 dan *Data Flow Diagram (DFD)* level 1 dapat dilihat pada Gambar 4.3 dan keterangannya dapat dilihat pada Tabel 4.8. Untuk penguraian *Data Flow Diagram* level 2 dan seterusnya dapat dilihat pada Lampiran A. Selain pada analisa fungsional sistem ini juga menjelaskan tentang *entity relationship diagram (ERD)* yang dapat dilihat pada Gambar 4.3.

##### 4.1.5.1. Context Diagram

Gambaran secara umum sistem rekomendasi buku perpustakaan dapat dilihat pada gambar *context diagram* dibawah ini



Gambar 4.3. *Context Diagram* Sistem Rekomendasi Buku Perpustakaan

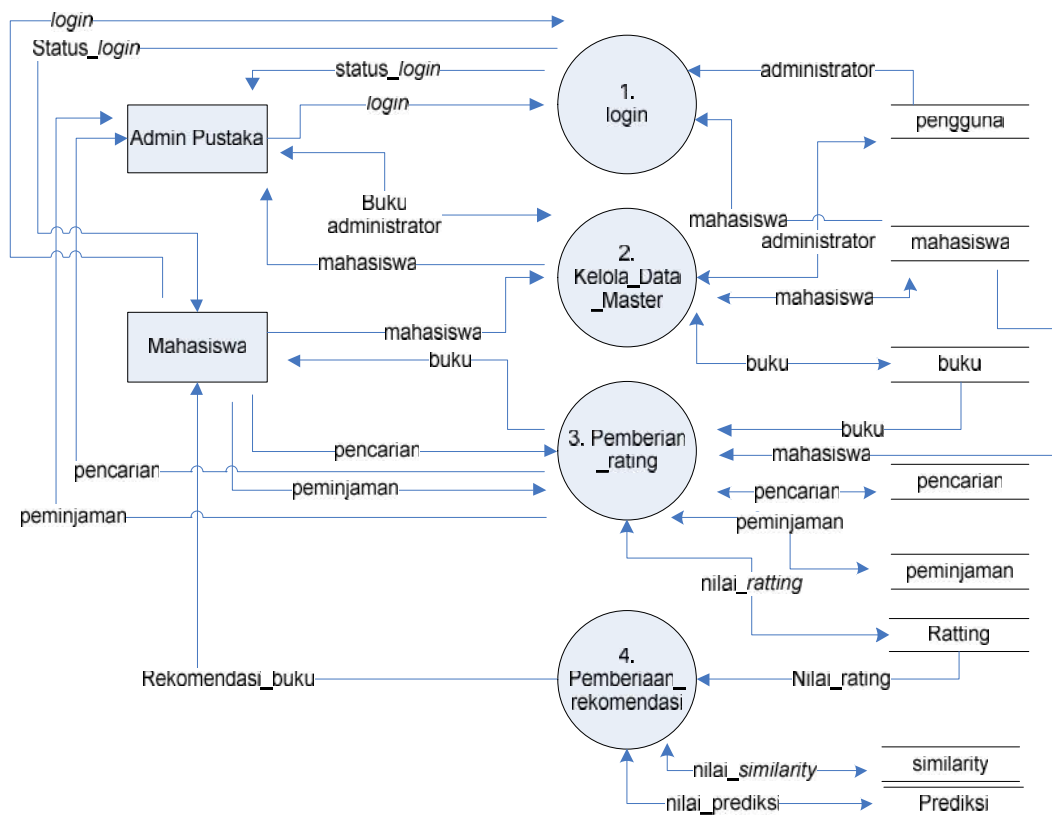
Tabel 4.4 Tabel *Context Diagram* Sistem Rekomendasi Buku Perpustakaan

No	Jenis	Nama	Deskripsi
1.	Entitas	Admin Pustaka	<i>Administrator</i> berperan sebagai pengelola sistem
		Mahasiswa	Menggunakan sistem untuk mendapatkan rekomendasi buku
2.	Aliran Data	<i>Administrator</i>	Data admin yang dapat <i>login</i> sebagai pengelola
		Buku	Kumpulan buku yang akan direkomendasi
		<i>login</i>	<i>Username</i> dan <i>password</i> yang digunakan untuk <i>login</i>
		Peminjaman	Aktifitas peminjaman mahasiswa yang tercatat kedalam database sistem
		Pencarian	Aktifitas pencarian buku yang

			dilakukan mahasiswa yang tercatat kedalam database sistem
		Mahasiswa	pengguna yang dapat memperoleh rekomendasi buku berdasarkan aktifitas peminjaman dan pencarian yang telah ia lakukan
		Status_login	informasi yang diperoleh setelah melakukan login.

#### 4.1.5.2. Data Flow Diagram

DFD akan menggambarkan sistem rekomendasi buku perpustakaan dengan lebih terperinci. Pada DFD terdapat proses-proses yang terjadi pada sistem rekomendasi buku seperti proses *login*, pengelolaan data master, pencarian , peminjaman dan proses pemberian rekomendasi buku. *Data flow diagram (DFD) level 1* dapat dilihat pada Gambar 4.4.



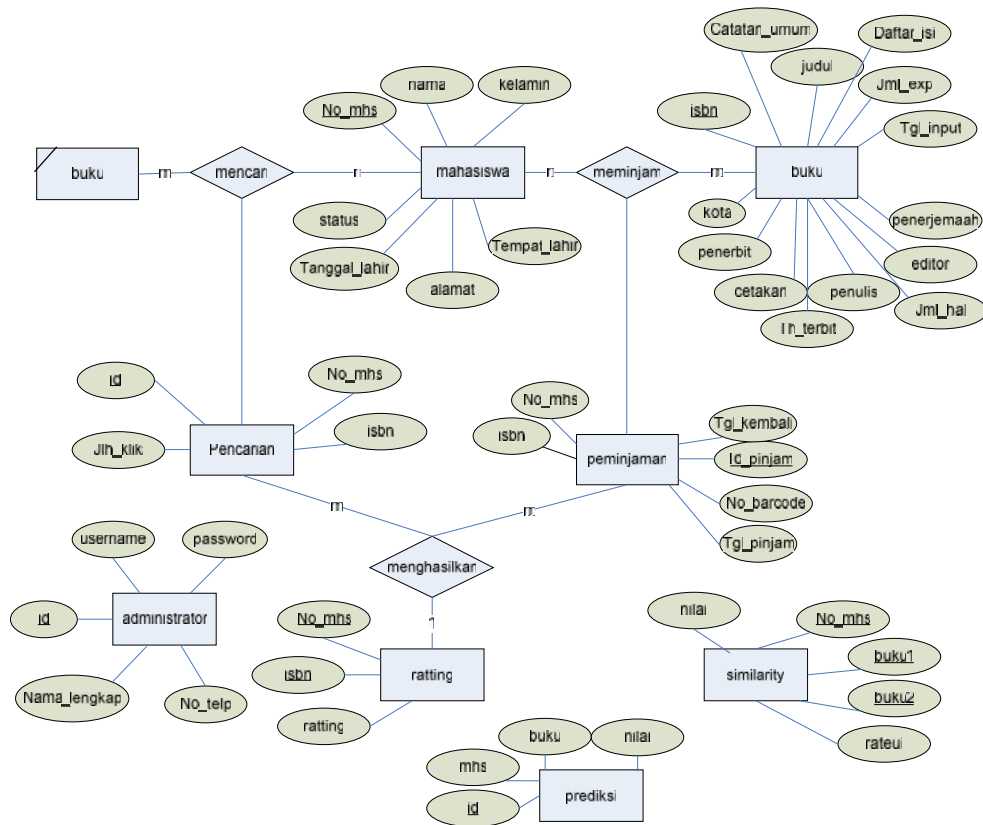
Gambar 4.4. DFD level 1 Sistem Rekomendasi Buku Perpustakaan

Tabel 4.5 Tabel DFD *level 1* Sistem Rekomendasi

No	Jenis	Nama	Deskripsi
1.	Proses	<i>Login</i>	Verifikasi <i>user</i> yang menggunakan sistem
		Kelola_Data_Master	Proses pengelolaan data master seperti data mahasiswa, buku, dan <i>administrator</i>
		Pemberian <i>rating</i>	Proses pemberian <i>rating</i> yang di peroleh melalui pencarian dan peminjaman yang dilakukan <i>user</i>
		Pemberian_rekomendasi	Proses penghitungan sampai pemberian rekomendasi buku kepada mahasiswa
2.	Aliran Data	<i>Administrator</i>	Data admin yang dapat <i>login</i> sebagai pengelola
		Buku	Kumpulan buku yang akan direkomendasi
		<i>Login</i>	<i>Username</i> dan <i>password</i> yang digunakan untuk <i>login</i>
		Peminjaman	Aktifitas peminjaman mahasiswa yang tercatat kedalam database sistem
		Pencarian	Aktifitas pencarian buku yang dilakukan mahasiswa yang tercatat kedalam database sistem
		Nilai_ <i>rating</i>	Nilai <i>rating user</i> terhadap buku yang pernah ia pinjam/cari
		Nilai_ <i>similarity</i>	Nilai kemiripan antar buku
		Nilai_ <i>prediksi</i>	Prediksi <i>rating user</i> yang sedang <i>login</i> terhadap buku yang akan direkomendasikan.

#### 4.1.5.3. Entity Relationship Diagram

Pada analisa data sistem akan dijelaskan mengenai hubungan antara data 1 dan data lainnyayang dapat dilihat pada Gambar 4.3. Pada ERD dapat dilihat terdapat 8 tabel dimana tabel buku berelasi dengan tabel mahasiswa yang akan menghasilkan tabel peminjaman dan tabel pencarian. Kedua tabel ini akan di konversikan menjadi nilai *rating* yang disimpan kedalam tabel *rating*. Untuk tabel *similarity* dan prediksi terdapat relasi dan tabel ini bersifat *temporary* (sementara).



Gambar 4.5. ERD Sistem Rekomendasi

## 4.2. Perancangan Sistem

Perancangan sistem dikelompokkan menjadi dua sub pokok bahasan, yaitu perancangan tabel, dan perancangan antarmuka.

### 4.2.1. Perancangan Tabel

- Tabel Administrator, digunakan untuk menampung data administrator sistem.

Tabel 4.6 Tabel Administrator

Nama Field	Type dan Lenght	Primary key	Null	Default
Id	integer(11)	Yes	No	-
Username	Varchar(30)	Unique	No	-
Password	Varchar(30)		No	-
Nama_lengkap	Varchar(30)		No	-
No_telp	Varchar(12)			-

- b. Tabel Mahasiswa, digunakan untuk menampung data mahasiswa yang dapat memperoleh rekomendasi buku.

Tabel 4.7 Tabel Mahasiswa

Nama Field	Type dan Lenght	Primary key	Null	Default
No_mhs	Varchar(20)	Yes	No	-
nama	Varchar(70)		No	-
status	Varchar(1)		No	-
alamat	Varchar(50)		No	-
Kelamin	Varchar(1)		No	-
Tempat_lahir	Varchar(50)		No	-
Tanggal_lahir	Varchar(10)		No	-

- c. Tabel Buku, digunakan untuk menampung data buku yang akan direkomendasikan.

Tabel 4.8 Tabel Buku

Nama Field	Type dan Lenght	Primary key	Null	Default
Isbn	Varchar(15)	Yes	No	-
Judul	Varchar(200)		No	-
Jml_exp	Varchar(10)		No	-
Tgl_input	Varchar(25)		No	-
Penerjemaah	Varchar(70)		No	-
Editor	Varchar(70)		No	-
Penulis1	Varchar(70)		No	-
Penulis2	Varchar(70)		No	-
Penulis3	Varchar(70)		No	-
Penulis4	Varchar(70)		No	-
Cetakan	Varchar(60)		No	-
Th_terbit	Varchar(10)		No	-
Kota	Varchar(60)		No	-
Jml_hal	Varchar(10)		No	-
Pj_lebar	Varchar(50)		No	-
Subyek	Varchar(100)		No	-
Daftarisi	Text		No	-
Catatan_umum	Varchar(100)		No	-

- d. Tabel peminjaman, digunakan untuk menampung data peminjaman mahasiswa yang digunakan sebagai nilai *rating*.



Tabel 4.9 Tabel Peminjaman

Nama Field	Type dan Lenght	Primary key	Null	Default
Id_pinjam	<i>integer(11)</i>	<i>Yes</i>	<i>No</i>	-
Isbn	<i>Varchar(15)</i>		<i>No</i>	-
No_mhs	<i>Varchar(20)</i>		<i>No</i>	-
Tgl_pinjam	<i>Varchar(10)</i>		<i>No</i>	-
No_barcode	<i>Varchar(50)</i>		<i>No</i>	-
Tgl_kembali	<i>Varchar(10)</i>		<i>No</i>	-

- e. Tabel pencarian, digunakan untuk menampung data aktifitas pencarian yang dilakukan mahasiswa yang digunakan sebagai nilai *rating*.

Tabel 4.10 Tabel Pencarian

Nama Field	Type dan Lenght	Primary key	Null	Default
Id	<i>integer(11)</i>	<i>Yes</i>	<i>No</i>	-
No_mhs	<i>Varchar(20)</i>		<i>No</i>	-
Isbn	<i>Varchar(15)</i>		<i>No</i>	-
Jlh_klik	<i>int(11)</i>		<i>No</i>	-

- f. Tabel *rating*, digunakan untuk menampung data *rating* user pada item.

Tabel 4.11 Tabel *Rating*

Nama Field	Type dan Lenght	Primary key	Null	Default
Isbn	<i>Varchar(15)</i>	<i>Yes</i>	<i>No</i>	-
No_mhs	<i>Varchar(20)</i>	<i>Yes</i>	<i>No</i>	-
<i>rating</i>	<i>int(11)</i>		<i>No</i>	-

- g. Tabel *similarity*, digunakan untuk menampung hasil perhitungan kemiripan *adjusted cosine*. Tabel ini bersifat *temporary*, tabel ini akan dikosongkan ketika akan dilakukan perhitungan.

Tabel 4.12 Tabel *Similarity*

Nama Field	Type dan Lenght	Primary key	Null	Default
Nomhs	<i>Varchar(20)</i>	<i>Yes</i>	<i>No</i>	-
Buku1	<i>Varchar(9)</i>	<i>Yes</i>	<i>No</i>	-
Buku2	<i>Varchar(9)</i>	<i>Yes</i>	<i>No</i>	-
Rateui	<i>int(2)</i>		<i>No</i>	-
Nilai	<i>Varchar(5)</i>		<i>No</i>	-

- h. Tabel prediksi, digunakan untuk menampung data *item* yang akan direkomendasikan beserta nilai prediksi *rating*-nya. Tabel ini juga bersifat *temporary* seperti tabel *similarity*.

Tabel 4.13 Tabel Prediksi

Nama Field	Type dan Lenght	Primary key	Null	Default
Id	<i>int(11)</i>	<i>Yes</i>	<i>No</i>	-
Mhs	<i>Varchar(20)</i>		<i>No</i>	-
Isbn	<i>Varchar(15)</i>		<i>No</i>	-
Nilai	<i>Varchar(5)</i>		<i>No</i>	-

#### 4.2.2. Perancangan Antarmuka

Pada sub bab ini, hanya menjelaskan rancangan antarmuka pada proses-proses utama dari sistem rekomendasi buku perpustakaan seperti *form* pencarian, peminjaman dan hasil rekomendasi. Antarmuka yang dibangun adalah sebagai berikut:

##### 4.2.2.1. Tampilan Utama

Tampilan pada halaman utama ini adalah tampilan dari sistem saat *user* membuka sistem dan berhasil *login*. Pada tampilan utama ini terdapat sebuah *form* pencarian yang akan digunakan oleh pengguna untuk mencari buku dimana hasil pencarian yang di klik akan menjadi nilai *rating user*. (Lihat Gambar 4.6)

<b>HEADER</b>	
	<div style="border: 1px solid black; border-radius: 10px; width: 80%; margin: 0 auto; padding: 5px;"> <input style="width: 80%; border: none; border-bottom: 1px solid black;" type="text"/> <span style="border: 1px solid black; border-radius: 5px; padding: 2px 5px; margin-left: 5px;">cari</span> </div>
<b>FOOTER</b>	

Gambar 4.6. Antarmuka *Form* Utama

#### 4.2.2.2. Tampilan Hasil Pencarian

Pada tampilan ini akan diberikan beberapa buku hasil pencarian yang dilakukan oleh *user*. Ketika *user* mengklik *item* maka sistem akan menyimpan ke *database* sebagai nilai *rating*. (Lihat Gambar 4.7)

HEADER	
	<div>Hasil Pencarian (klik untuk detail):</div> <ul style="list-style-type: none"><li>- Buku 1</li><li>- Buku 2</li><li>- Buku 3</li><li>- Buku 4</li></ul>
FOOTER	

Gambar 4.7. Antarmuka Hasil Pencarian

#### 4.2.2.3. Tampilan Peminjaman

Pada tampilan ini akan ditampilkan detail buku dan terdapat tombol pinjam yang terletak dibagian bawah. Ketika *user* mengklik tombol tersebut maka sistem akan menyimpan ke *database* sebagai nilai *rating*. (Lihat Gambar 4.8)

HEADER	
	<div>Detai Buku :</div> <div>Judul : abc</div> <div>Pengarang : abc</div> <div>Penerbit : abc</div> <div>Tahun terbit : 2012</div> <div>pinjam</div>
FOOTER	

Gambar 4.8. Antarmuka Peminjaman

#### 4.2.2.4. Hasil Rekomendasi

Pada tampilan ini akan ditampilkan buku-buku yang akan direkomendasikan kepada *user*.(Lihat Gambar 4.9)

HEADER	
	<p>Buku rekomendasi :</p> <ul style="list-style-type: none"><li>- Buku 1</li><li>- Buku 2</li><li>- Buku 3</li><li>- Buku 4</li></ul>
FOOTER	

Gambar 4.9. Antarmuka Hasil Rekomendasi

#### 4.2.2.5. Halaman Utama Untuk *Administrator*

Antarmuka pada Gambar 4.10 di bawah ini merupakan antarmuka yang akan digunakan untuk setiap menu yang terdapat dalam halaman *administrator*.

HEADER	
MENU	
FOOTER	

Gambar 4.10. Antarmuka Halaman *Administrator*

#### 4.2.2.6. HalamanTampil Data Untuk *Administrator*

Antarmuka pada gambar di bawah ini merupakan antarmuka yang akan digunakan untuk melihat data seperti data administrator, buku, mahasiswa, peminjaman dan pencarian. Setiap data akan direpresentasikan dalam sebuah tabel.

HEADER									
	<table border="1"><thead><tr><th></th><th></th><th></th><th></th></tr></thead><tbody><tr><td></td><td></td><td></td><td></td></tr></tbody></table>								
FOOTER									

Gambar 4.11. Antarmuka Tampil Data

#### 4.2.2.7. HalamanTambah Data

Antarmuka pada Gambar 4.1 merupakan antarmuka form penambahan data kedalam *database*.

HEADER	
	<div>Form Tambah data</div> <div><div>inputan <input type="text"/></div><div>inputan <input type="text"/></div><div>inputan <input type="text"/></div><div><input type="button" value=""/></div></div>
	FOOTER

Gambar 4.12. Antarmuka Tambah Data

#### 4.2.2.8. Halaman *Import Data Excel*

Antarmuka pada gambar di bawah ini merupakan antarmuka untuk alternatif penambahan data ketika data yang *input*-kan dalam jumlah yang banyak. Data dihimpun kedalam sebuah file .xls kemudian file tersebut di-*import*-kan kedalam *database*.

HEADER	
	<div>Import data excel</div> <div><input type="text"/> <input type="button" value="Import"/></div>
FOOTER	

Gambar 4.13. Antarmuka Import Data Excel

## **BAB V**

### **IMPLEMENTASI DAN PENGUJIAN**

Pada bab implementasi dan pengujian ini pembahasan akan dibagi menjadi dua bagian, yaitu pembahasan tentang implementasi dan pembahasan tentang pengujian.

#### **5.1. Implementasi**

Implementasi merupakan tahap yang siap dioperasikan pada keadaan yang sebenarnya, sehingga akan diketahui apakah sistem yang dibuat benar-benar dapat menghasilkan tujuan yang ingin dicapai atau tidak, serta apakah sistem dapat digunakan sebagaimana mestinya sesuai dengan analisa dan perancangan yang telah dibahas sebelumnya.

##### **5.1.1. Batasan Implementasi**

Batasan implementasi dari Tugas Akhir ini adalah:

1. Menggunakan bahasa pemograman PHP dan *database* MySQL.
2. *Data* yang digunakan adalah data buku yang terdapat di perpustakaan Fakultas Sains dan Teknologi, tidak termasuk laporan kerja praktek (KP) dan laporan tugas akhir (TA).

##### **5.1.2. Lingkungan Operasional**

Dalam menjalankan sistem atau aplikasi terdapat beberapa sarana pendukung yaitu berupa peralatan-peralatan yang sangat berperan dalam menunjang penerapan sistem yang didesain terhadap pengolahan data. Diantaranya adalah lingkungan Perangkat Lunak (*Software*) dan lingkungan Perangkat keras (*Hardware*).

Lingkungan operasional merupakan lingkungan dimana sistem ini digunakan. Spesifikasi kebutuhan dari lingkungan operasional, yaitu:

1. Perangkat keras

Processor : *Intel Core i5 2.4 GHz*

Memori (RAM) : 2.00 GB

2. Perangkat Lunak

Sistem Operasi : *Windows 7 Professional 64 Bit*

Bahasa Pemrograman : *PHP version 5.3.1*

Browser : *Google Chrome version 20.0.1132.47*

DBMS : *MySQL version 5.1.41*

Web Server : *Apache version 2.2.14*

Tools : *Notepad ++ v5.9.3*

### 5.1.3. Hasil Implementasi

Setelah tahap analisa dan perancangan selesai dilakukan, maka dilanjutkan dengan tahap implementasi sistem dari hasil analisa yang telah diperoleh dan mengimplementasikan hasil perancangan *interface* yang telah dibuat. Berikut ini akan dijelaskan mengenai hasil implementasi dari rancang bangun sistem rekomendasi buku perpustakaan, dimana pada sistem tersebut memiliki lima menu, yaitu menu Beranda, Buku Rekomendasi, Buku Terbaru, Petunjuk, Tentang Kami. Menu Beranda merupakan menu yang akan menampilkan halaman utama sistem. Pada menu ini terdapat *form* pencarian buku agar memudahkan *user* mencari buku yang diinginkan. Menu Buku Rekomendasi berisi buku-buku yang akan direkomendasikan. Menu buku terbaru berisi tentang 100 yang baru saja di inputkan. Menu petunjuk berguna untuk memberikan petunjuk kepada *user* tentang bagaimana menggunakan sistem rekomendasi. Tentang kami berisi tentang profil penulis.

Berikut implementasi *interface* sistem rekomendasi buku sesuai dengan menu yang ada pada sistem:



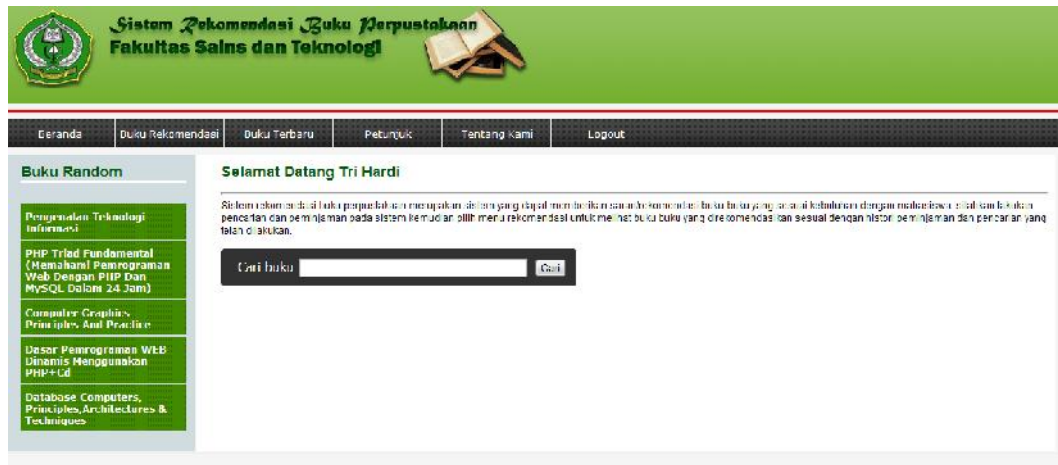
## 1. Implementasi Tampilan *Login*



Gambar 5.1. Tampilan Menu *Login*

Agar dapat mengakses system, *user* terlebih dahulu harus *login* agar sistem dapat melakukan perhitungan berdasarkan *history* peminjaman yang dilakukan *user*.

## 2. Implementasi Tampilan Menu Beranda



Gambar 5.2. Tampilan Menu *Home*

Tampilan pada Gambar 5.2 merupakan tampilan yang akan muncul pertama sekali ketika *user* menjalankan sistem ini. Setelah melakukan pencarian pada *form* yang disediakan *user* akan dibawa kehalaman hasil pencarian, dimana halaman ini berisi buku-buku yang sesuai dengan *keyword* pencarian *user*. Gambar 5.3 dibawah ini menunjukkan tampilan hasil pencarian.



Gambar 5.3. Hasil Tampilan Hasil Pencarian

Buku hasil pencarian tersebut dapat diklik *user* dimana jumlah klik *user* pada buku inilah yang nantinya akan menjadi penilaian *rating*. Setelah melakukan klik, *user* akan dibawa ke tampilan detail buku. Pada tampilan ini diterangkan mengenai pengarang, tahun terbit, edisi, daftar isi dan informasi lain menai buku tersebut. Gambar 5.4 menunjukkan tampilan detail buku.



Gambar 5.4. Tampilan Menu Detail Buku

### 3. Implementasi Tampilan Menu Buku Rekomendasi



**Buku Rekomendasi**

Buku yang direkomendasi untuk Anda yaitu:

NO	JUDUL	NILAI PREDIKSI
1	Intellectual Social Energy	6
2	Komputer dan Masyarakat	6
3	Struktur Matematika dan Logika: ARSIT TITIK DAN RUMUS	6
4	Dasar-dasar Pengembangan dan Implementasi Database Relasional	6
5	Pengantar Sistem Operasi komputer : Plus Studi Kasus Komputasi Linux	6
6	Derajat Manusia dan Komputer	-6
7	Ukuran dan Komputer Sistem Jelas Data	-6
8	Prinsip Pemrograman Java 2	6
9	Sistem Operasi	-1.00

Gambar 5.5. Tampilan Menu Buku Rekomendasi

Pada menu ini, buku yang merupakan hasil akhir/*output* dari sistem akan ditampilkan. *User* dapat melihat buku apa saja yang direkomendasikan untuknya.

### 4. Implementasi Menu Buku Terbaru



**Buku Terbaru**

- Pemrograman Java Dasar-dasar pemrograman dan Pemahaman
- Membangun Ilmu dengan Paradigma Islam
- Logika dan Algoritma Dasar menggunakan Bahasa C++
- 55 Kreasi populer Animasi cantik dengan Adobe Flash
- Renaisans sebuah Percikan
- Tip Jitu Optimasi Jaringan Wi-Fi
- Pemodelan Basis Data Berorientasi Objek
- Membuat sendiri Aplikasi Android untuk Pemula
- Sistem kontrol via Web dengan CGI,PHP,dan Ajax
- Just XML
- Pemrograman Game dengan Java dan GTGE
- Panduan Praktis merawat hard disk bagi semua orang
- Algoritma Pemrograman menggunakan C++
- Modul Pembelajaran Algoritma dan Pemrograman
- Cara Praktis Membius Antivirus Komputer
- Statistik Konsep Dasar & Aplikasinya
- Produktifitas Apa dan Bagaimana
- Sistem Informasi Manajemen
- Pake Flash bikin Animasi Teks paling keren
- Langkah Mudah Melakukan Analisis Statistik menggunakan SPSS 19
- Mengembangkan Aplikasi Enterprise berbasis Android
- Pengantar Manajemen
- Sistem Operasi
- Dasar Pemrograman web Dinamis Menggunakan PHP (Revisi)
- Using Information Technology (Pengertian Praktis Dunia Komputer Dan Komunikasi) Edisi 7
- Pengantar Komputer Numpak Dengan Melayah

Gambar 5.6. Tampilan Menu Buku Terbaru

Tampilan ini berisikan buku-buku yang baru saja di-*entry* kedalam sistem. *User* dapat melihat buku-buku terbaru melalui tampilan ini.

## 5. Impementasi Menu Petunjuk



Gambar 5.7. Tampilan Menu Petunjuk

Pada menu ini diinformasikan kepada *user* mengenai tata cara penggunaan sistem rekomendasi buku perpustakaan.

## 6. Implementasi Menu Tentang Kami



Gambar 5.8. Tampilan Menu Tentang Kami

Menu ini berisi sekilas tentang penulis yang merancang sistem rekomendasi buku perpustakaan dengan menggunakan metode *collaborative based filtering*.



## 7. Implementasi Halaman Admin

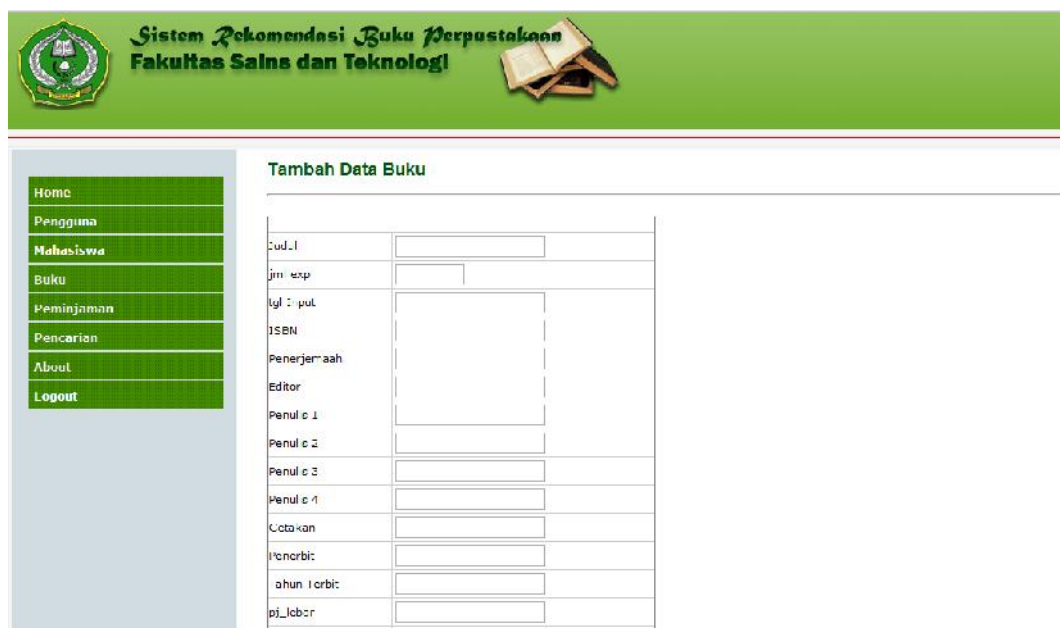


NO	ID PEMINJAMAN	BARCODE	JUDUL	TGL PINJAM	NO MAHASISWA	TGL KEMBALI	EDIT	DEL
1	735	7377	1861 Fuzzy Logic: Pendekatan Manual Fuzzy Membership Function	09/05/2012	075/FST/IT/2008			
2	794	7115	1856 Analysis Time Series: Penerapan dan Uji Statistik Mulus dan	09/05/2012	075/FST/IT/2008			
3	793	8505	2006 Rancangan dan Uji Perhitungan	09/05/2012	10551008133			
4	792	8802	1820 Pengantar Java Source Code	09/05/2012	11155201979			
5	791	8511	1402 Sistem Queue	09/05/2012	11155201979			
6	790	8081	658 Metode Numerik: Berbasis MATLAB + Disket	09/05/2012	10555005607			
7	789	1010	1250 Quality Control (+ Disket)	09/05/2012	10555005607			
8	788	6184	671 Theory and Application of IT Research: Metodologi Penelitian	09/05/2012	065/FST/IT/2009			
9	787	7074	658 Konsep Sistem Informasi Geografis	09/05/2012	11151201910			
10	786	1014	405 Introduction to Operations Research (1st ed.)	09/05/2012	016/FST/MT/2010			
11	785	0349	85 Applied Mathematical Methods with MATLAB for Engineers and	09/05/2012	016/FST/MT/2010			
12	784	8372	772 Peminjaman Web Menggunakan HTML, CSS, JavaScript & PHP	09/05/2012	144/FST/ST/2010			
13	783	7512	1875 Database: Pengantar dan Java	09/05/2012	044/FST/ST/2010			

Gambar 5.9. Tampilan Halaman Admin

Interface pada Gambar 5.9 merupakan tampilan untuk halaman *administrator*. Pada halaman ini admin dapat mengolah data buku, mahasiswa, pencarian, peminjaman, *rating* dan lain lain. Admin dapat melakukan penambahan data, penghapusan dan *update* data.

## 8. Implementasi Tampilan Penambahan Data Pada Halaman Admin



**Tambah Data Buku**

Judul:

Jumlah:

Upt:

ISBN:

Penerjemah:

Editor:

Penulis 1:

Penulis 2:

Penulis 3:

Penulis 4:

Cetakan:

Nomorbit:

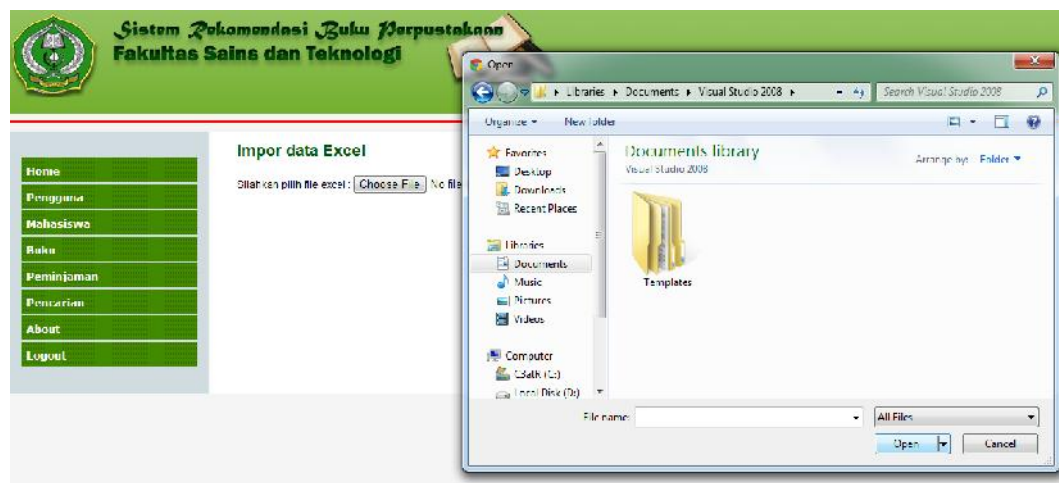
dan tahun terbit:

p\_j\_lobr:

Gambar 5.10. Tampilan Halaman Tambah Data

Tampilan pada Gambar 5.10 merupakan tampilan *form* pengisian data. Admin dapat memasukkan data ke sistem secara langsung dengan mengisi *form-form* yang telah disediakan berdasarkan menu masing masing. Pengecualian untuk menu pencarian, admin tidak dapat melakukan penambahan data secara langsung karena data ini diperoleh melalui aktifitas *user* pada system.

Selain menginputkan secara langsung, admin juga dapat memasukkan data secara masal dengan menggunakan fasilitas *import* dari *excel*. Admin memasukkan data berupa dokumen *excel* kemudian data tersebut dikonversikan kedalam data *mysql* sehingga dapat diolah. Gambar 5.8 menunjukkan tampilan menu *import* data dari *excel*.



Gambar 5.11. Tampilan Halaman *Import* Data *excel*

## 9. Implementasi Tampilan Konversi *Rating*

Sebelum dilakukan perhitungan kemiripan, terlebih dahulu data peminjaman dan pencarian dikonversikan menjadi data *rating*. Proses pengkonversian data pencarian dan peminjaman ini terletak dihalaman admin. Proses ini hanya perlu dilakukan sekali ketika data baru diinputkan. Apabila setelah data-rata *rating* diinputkan terjadi proses peminjaman atau pencarian, sistem secara otomatis akan meng-*update* data *rating* tersebut. Berikut ini merupakan tampilan ketika proses konversi data peminjaman menjadi data *rating*.



## **5.2. Pengujian**

Setelah tahap implementasi selesai, maka dilanjutkan dengan pengujian dari implementasi yang telah dibuat. Pengujian sistem dilakukan dengan tujuan untuk menjamin sistem yang dibangun sesuai dengan hasil analisa dan perancangan sehingga dapat dibuat satu kesimpulan akhir. Pengujian fungsionalitas sistem dilakukan dengan metode *blackbox*. Selain itu, pengujian juga dilakukan dengan melihat MAE (*Mean Absolute Error*) pada prediksi *rating* yang diberikan dan nilai presisi dari hasil UAT.

### **5.2.1. Pengujian *Blackbox* pada Sistem Rekomendasi Buku Perpustakaan**

Pengujian sistem dilakukan dengan menggunakan metode *Blackbox*, yaitu metode pengujian yang berfokus pada kebutuhan fungsional perangkat lunak. Upaya pengujian dimaksudkan untuk memastikan bahwa fungsi-fungsi yang ada di dalam perangkat lunak dapat berjalan dengan baik, dan menginisialisasi kesalahan fungsi untuk kemudian dapat diperbaiki. Adapun hasil dari pengujian ini dapat dilihat pada Tabel 5.1.



Tabel 5.1. Hasil Pengujian Sistem dengan Metode *Blackbox*

No.	Nama Pengujian	Deskripsi	Prekondisi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
1.	Verifikasi	Pengujian verifikasi login dengan memasukan <i>username</i> dan <i>password</i> yang benar untuk <i>level administrator</i> .	<i>Form login</i> telah tampil	Masukkan <i>username</i> dan <i>password</i>	Masukkan <i>username</i> “admin” <i>Password</i> “admin”	Tampil halaman administrator.	Tampil halaman administrator.	Benar
		Pengujian verifikasi login dengan memasukan <i>username</i> dan <i>password</i> yang benar untuk <i>level mahasiswa</i> .			Masukkan <i>username</i> “11151100118” <i>Password</i> “11151100118”	Tampil halaman utama mahasiswa.	Tampil halaman utama mahasiswa.	Benar
		Pengujian verifikasi login dengan memasukan <i>username</i> dan <i>password</i> yang salah.			Masukkan <i>username</i> “11151100118” <i>Password</i> “masuk”	Muncul peringatan “Maaf <i>username</i> dan <i>password</i> anda salah” Lalu kembali ke <i>form login</i>	Muncul peringatan “Maaf <i>username</i> dan <i>password</i> anda salah” Lalu kembali ke <i>form login</i>	Benar
2.	Pengujian <i>rating</i> Pencarian	Pengujian pemberian <i>rating</i> ketika <i>user</i> melakukan 1 kali klik pada buku yang dicari	Buku belum pernah diklik oleh <i>user active</i>	Lakukan klik pada buku “PHP & PostegreSQL”	<i>User</i> “11151100118” Buku “PHP & PostegreSQL”	Nilai <i>rating</i> 1 untuk buku “PHP & PostegreSQL”	Nilai <i>rating</i> 1 untuk buku “PHP & PostegreSQL”	Benar
		Pengujian pemberian <i>rating</i> ketika <i>user</i>	Buku telah diklik 1 kali	Lakukan klik pada buku	<i>User</i> “11151100118”	Nilai <i>rating</i> 2 untuk buku	Nilai <i>rating</i> 2 untuk buku	Benar

		melakukan 2 kali klik pada link pencarian		“PHP & PostegreSQL”	Buku “PHP & PostegreSQL”	“PHP & PostegreSQL”	“PHP & PostegreSQL”	
		Pengujian pemberian <i>rating</i> ketika <i>user</i> melakukan lebih dari 5 kali klik pada buku yang dicari	Buku telah diklik 6 kali	Lakukan klik pada buku “Database Relasional dengan MySQL”	<i>User</i> “11151100118” Buku “Database Relasional dengan MySQL”	Nilai <i>rating</i> 5 untuk buku “Database Relasional dengan MySQL”	Nilai <i>rating</i> 5 untuk buku “Database Relasional dengan MySQL”	Benar
3	Pengujian <i>rating</i> peminjaman	Pengujian pemberian <i>rating</i> ketika <i>user</i> melakukan 1 kali peminjaman	Data peminjaman telah masuk ke <i>database</i> , Buku belum pernah dipinjam <i>user active</i>	Klik link konversi peminjaman pada halaman administrator	<i>User</i> “11151100118” Buku “Konsep Kecerdasan Buatan”	Nilai <i>rating</i> 6 untuk buku “Konsep Kecerdasan Buatan”	Nilai <i>rating</i> 6 untuk buku “Konsep Kecerdasan Buatan”	Benar
		Pengujian pemberian <i>rating</i> ketika <i>user</i> melakukan 2 kali peminjaman	Data peminjaman telah masuk ke <i>database</i> , Buku 1 kali dipinjam <i>user active</i>		<i>User</i> “11151100118” Buku “Teknik Digital”	Nilai <i>rating</i> 7 untuk buku “Teknik Digital”	Nilai <i>rating</i> 7 untuk buku “Teknik Digital”	Benar
		Pengujian pemberian <i>rating</i> ketika <i>user</i> melakukan lebih dari 5 kali klik peminjaman	Data peminjaman telah masuk ke <i>database</i> , Buku 5 kali dipinjam <i>user</i>		<i>User</i> “11151100118” Buku “Algoritma Pemrograman”	Nilai <i>rating</i> 10 untuk buku “Algoritma Pemrograman”	Nilai <i>rating</i> 10 untuk buku “Algoritma Pemrograman”	Benar

### 5.2.2. Pengujian Kualitas Rekomendasi

Pengujian kualitas rekomendasi dilakukan dengan 2 cara yaitu menggunakan *User Acceptance Test* (UAT) dan fungsi perhitungan *error*. Pada pengujian ini diambil *sample* data sebanyak 650 buku dan jumlah data *rating* yang terhimpun sebanyak 841 *instance*. Untuk menguji tingkat akurasi dilakukan dengan menghitung nilai *error* pada prediksi yang dihasilkan. Persamaan yang digunakan untuk menghitung nilai *error* yaitu persamaan MAE (2.8).

#### 5.2.2.1. Prosedur Pengujian Kualitas Rekomendasi

Prosedur pengujian kualitas rekomendasi dibedakan berdasarkan parameter berikut :

1. UAT (*user acceptance test*)

Pada pengujian UAT diberikan kuisioner kepada mahasiswa yang pernah meminjam atau mencari beberapa buku perpustakaan. Pada kuisioner UAT untuk mahasiswa berisi *list* buku yang pernah dipinjam dan *list* buku yang direkomendasikan sistem untuk *user*. *User* akan menilai buku-buku yang sesuai dengan minat/kebutuhan mereka pada *list* buku yang direkomendasikan sistem.

Kuisioner dibedakan berdasarkan responden yang mengisi kuisioner tersebut yaitu petugas perpustakaan dan mahasiswa. Jumlah pertanyaan yang diajukan untuk petugas perpustakaan adalah 7 pertanyaan, sedangkan untuk mahasiswa cukup menilai buku-buku hasil rekomendasi sistem.

2. Tingkat akurasi berdasarkan fungsi perhitungan *error*

Tahap awal dari pengujian tingkat akurasi yaitu membagi *dataset* menjadi dua, dimana 70% untuk data latih dan 30% sisanya sebagai data uji. Parameter pengujiannya berdasarkan tingkat *sparsity* (kekosongan data *rating*) karena masalah utama pada metode *collaborative filtering recommender system* yaitu tingkat *sparsity* yang tinggi. Pada parameter uji

tersebut akan dilakukan percobaan sebanyak 5 kali. Persamaan yang digunakan untuk menghitung tingkat akurasi adalah persamaan 2.8.

Pada pengujian data *sample*, akan dilakukan pemilihan secara acak data *rating* untuk dikosongkan dengan tingkat *sparsity* sebanyak 30% 50% dan 60%. Karena masalah *sparsity* merupakan masalah yang kerap kali terjadi pada metode *collaborative filtering*, maka dari itu melalui pengujian ini akan dilihat seberapa besar pengaruh kekosongan nilai *rating* terhadap nilai akurasi prediksi sistem.

Perhitungan nilai *error* pada MAE dilakukan dengan menghitung rata-rata selisih antara nilai *rating* asli dengan *rating* prediksi. Misalnya pada *dataset* terdapat 5 *item* yang di-*rating*. Untuk setiap *item*, nilai *rating* prediksi yang diberikan sistem dikurangi dengan nilai *rating* asli yang diperoleh dari *user*, selisih *rating* asli dan *rating* prediksi dijumlahkan dan dibagi banyak *item* (5 *item*).

#### **5.2.2.2. Hasil Pengujian Kualitas Rekomendasi**

##### **1. UAT (*user acceptance test*)**

Hasil dari *user acceptance test* dengan cara pengisian angket menjelaskan apakah sistem yang dibangun layak atau tidak dalam pemberian rekomendasi buku perpustakaan. Untuk kuisioner yang ditujukan kepada petugas pustaka dan kuisioner mahasiswa secara keseluruhan dapat dilihat pada Lampiran B.

Pada Tabel 5.2 Berikut ini adalah hasil jawaban angket atau kuisioner yang telah disebarakan kepada 6 mahasiswa yang telah mencoba sistem yang dibangun:

Tabel 5.2. Jawaban Hasil Pengujian dengan Kuisisioner Kepada Mahasiswa

No	Mahasiswa	Jumlah Buku Yang Dipinjam	Jumlah Buku Yang Direkomendasi	Jumlah Buku Yang Relevan	Jumlah Buku Yang Tidak Relevan	<i>Precision</i>
1.	Husni Wardi	5	20	6	14	0,3
2.	Indah Inzani Septa	16	33	8	25	0,24
3.	Isma Juliani	7	16	8	8	0,5
4.	Nur Alfi Laily	9	29	11	18	0,37
5.	Lia Septi Lestari	12	28	8	20	0,28
6.	Frita Yola Anggina	12	26	4	22	0,15
Rata-rata						0.306667

Dari hasil angket yang telah disebarkan kepada mahasiswa menghasilkan kesimpulan bahwa nilai rata-rata *precision* untuk buku yang direkomendasikan sistem kepada *user* rata-rata 0.30667. Dari hasil nilai *precision* tersebut dapat dikatakan bahwa sistem rekomendasi buku perpustakaan dengan metode *collaborative filtering* memberikan hasil rekomendasi buku yang relevan dengan *user* sebanyak 30%.

Pada penelitian ini peminjaman yang dilakukan *user* pada waktu yang lalu dengan peminjaman yang terkini tidak dibedakan. Sehingga buku-buku yang pernah disukai *user* pada waktu lalu ikut direkomendasikan walaupun buku tersebut sudah tidak diminati atau dibutuhkan *user*. Selain itu jumlah buku yang terbatas mengakibatkan tidak ada lagi buku yang relevan yang dapat direkomendasikan. Seperti kasus yang terjadi pada *user* Indah Inzani Septa yang memiliki minat pada buku matlab. Tetapi karena buku matlab yang terdapat di perpustakaan sebagian besar telah dipinjam atau dicari, hasil rekomendasi tidak memberikan rekomendasi buku matlab.

## 2. Tingkat akurasi berdasarkan fungsi perhitungan *error*

Pengujian tingkat akurasi yaitu menghitung tingkat *error* hasil rekomendasi yang diberikan sistem dengan menggunakan persamaan 2.8 dimana data yang diuji adalah data uji yang nilai *rating*-nya dikosongkan dengan tingkat *sparsity* tertentu.

Data akan diujikan pada 6 *user* yang diambil secara acak. *User* yang digunakan untuk pengujian yaitu : Agus (11053101853), Indah (079/FST/TIF/2008), Aritha (024/FST/TIF/2008), Khairu (090/P.FST/TIF.08/201), Syaputra (11153101866), Ary (11151100118),. Dimana untuk masing-masing *user* diketahui bahwa *user* Agus me-*rating* sebanyak 10 buku, *user* Indah me-*rating* sebanyak 16 buku, *user* Aritha me-*rating* sebanyak 60 buku, *user* Khairu me-*rating* sebanyak 15 buku, *user* Syaputra me-*rating* sebanyak 34 buku, *user* Ary me-*rating* sebanyak 75 buku,

Nilai MAE diperoleh dengan membandingkan *rating* yang diberikan *user* dengan *rating* yang diberikan sistem. Berikut ini contoh perhitungan nilai MAE untuk hasil rekomendasi buku yang diberikan kepada *user* Agus.

Pada saat tingkat *sparsity* 30% untuk buku pertama diperoleh nilai *rating* prediksi 9 dan *rating* asli 8. Buku ke 2 nilai *rating* prediksi 8 dan *rating* asli 9. Buku ke 3 nilai *rating* prediksi 6.55118 dan *rating* asli 6. Buku ke 4 nilai *rating* prediksi 6.51922 dan *rating* asli 6. Buku ke 5 nilai *rating* prediksi 6 dan *rating* asli 7. Buku ke 6 nilai *rating* prediksi 6 dan *rating* asli 7.

$$\begin{aligned} \text{MAE} &= \frac{\sum_{i=1}^n |p_i - q_i|}{N} \\ &= \frac{|9-8| + |8-9| + |6.55118-6| + |6.51922-6| + |6-7| + |6-7|}{6} \\ &= \frac{4.84267}{6} \\ &= 0.845067 \end{aligned}$$

Pada saat tingkat *sparsity* 50% untuk buku pertama diperoleh nilai *rating* prediksi 9 dan *rating* asli 8. Buku ke 2 nilai *rating* prediksi 8 dan

*rating* asli 9. Buku ke 3 nilai *rating* prediksi 6.54441 dan *rating* asli 6. Buku ke 4 nilai *rating* prediksi 6.51391 dan *rating* asli 6. Buku ke 5 nilai *rating* prediksi 6 dan *rating* asli 7. Buku ke 6 nilai *rating* prediksi 6 dan *rating* asli 7.

$$\begin{aligned} \text{MAE} &= \frac{|9-8|+|8-9|+|6.54441-6|+|6.51391-6|+|6-7|+|6-7|}{6} \\ &= \frac{5.05832}{6} \\ &= 0.843053 \end{aligned}$$

Pada saat tingkat *sparsity* 60% untuk buku pertama diperoleh nilai *rating* prediksi 9 dan *rating* asli 8. Buku ke 2 nilai *rating* prediksi 8 dan *rating* asli 9. Buku ke 3 nilai *rating* prediksi 6 dan *rating* asli 1. Buku ke 4 nilai *rating* prediksi 6 dan *rating* asli 6. Buku ke 5 nilai *rating* prediksi 6 dan *rating* asli 7. Buku ke 6 nilai *rating* prediksi 4.92695 dan *rating* asli 6.

$$\begin{aligned} \text{MAE} &= \frac{|9-8|+|8-9|+|6-1|+|6-6|+|6-7|+|4.92695-6|}{6} \\ &= \frac{9.07305}{6} \\ &= 1.512175 \end{aligned}$$

Pada saat tingkat *sparsity* 63% untuk buku pertama diperoleh nilai *rating* prediksi 9 dan *rating* asli 8. Buku ke 2 nilai *rating* prediksi 8 dan *rating* asli 9. Buku ke 3 nilai *rating* prediksi 6 dan *rating* asli 6. Buku ke 4 nilai *rating* prediksi 6 dan *rating* asli 6. Buku ke 5 nilai *rating* prediksi 6 dan *rating* asli 1. Buku ke 6 nilai *rating* prediksi 4.9095 dan *rating* asli 6.

$$\begin{aligned} \text{MAE} &= \frac{|9-8|+|8-9|+|6-6|+|6-7|+|6-1|+|4.9095-6|}{6} \\ &= \frac{9.0905}{6} \\ &= 1.5150833 \end{aligned}$$

Pada saat tingkat *sparsity* 65% untuk buku pertama diperoleh nilai *rating* prediksi 9 dan *rating* asli 8. Buku ke 2 nilai *rating* prediksi 8 dan *rating* asli 9. Buku ke 3 nilai *rating* prediksi 6 dan *rating* asli 7. Buku ke 4 nilai *rating* prediksi 6 dan *rating* asli 1. Buku ke 5 nilai *rating* prediksi 4.35904 dan *rating* asli 6.

$$\begin{aligned}
 \text{MAE} &= \frac{|9-8|+|8-9|+|6-7|+|6-1|+|4.35904-6|}{5} \\
 &= \frac{9.64096}{5} \\
 &= 1.928192
 \end{aligned}$$

Setelah dilakukan pengujian terhadap *user* dengan tingkat *sparsity* yang berbeda, maka didapatkan hasil tingkat akurasi prediksi untuk setiap tingkatan *sparsity* diperoleh nilai MAE sebagai berikut :

Tabel 5.3. Nilai MAE Berdasarkan Tingkat *Sparsity*

Tingkat <i>Sparsity</i>	Percobaan pada <i>user</i> ke -						<i>Rata-rata</i>
	I	II	III	IV	V	VI	
30%	0.84506	0.7717	0.5275	0.4191	0.8915	0.8392	0.715677
50%	0.84305	0.7711	0.5816	0.40695	0.90781	0.8733	0.753503
60%	1.5121	0.79454	0.6602	0.86287	0.7681	0.70803	0.884307
63%	1.515	0.80482	0.68042	0.8652	0.82798	0.7397	0.90552
65%	1.9281	0.77129	0.7763	0.7205	0.91777	0.8559	0.994977

Pada Tabel 5.3 dapat dilihat hasil pengujian *dataset* yang dilakukan pada data uji dengan tingkat *sparsity* sebesar 30% 50% 60%.63% dan 65% Dari hasil pengujian dapat diketahui bahwa nilai rata-rata MAE yang terendah adalah 0.715677 pada tingkat *sparsity* 30% sedangkan nilai rata-rata MAE yang tertinggi adalah 0.994977 pada tingkat *sparsity* 65%.

Hal ini menunjukkan bahwa pada pengujian tingkat *sparsity* dimana nilai MAE akan meningkat ketika tingkat kekosongan data dinaikkan. Peningkatan ini disebabkan karena ketika data dalam keadaan *sparse* yang tinggi jumlah data *rating* yang berkurang akan mempengaruhi proses pertimbangan dalam pemberian prediksi.

### 5.2.3. Pengujian Waktu Eksekusi

Pengujian waktu eksekusi dilakukan dengan menghitung waktu yang digunakan untuk memproses *dataset*. Pada pengujian ini akan dilihat pengaruh



banyak *dataset* dan pengaruh tingkat *sparsity* dengan waktu eksekusinya. Total data buku yang digunakan sebanyak 4000 data. Sedangkan untuk tingkat *sparsity dataset* yang digunakan adalah data uji yang telah ditentukan pada proses sebelumnya.

#### 5.2.3.1. Prosedur Pengujian Waktu Eksekusi

Pada pengujian waktu eksekusi *dataset* dipilih secara acak dengan jumlah tertentu. Pengujian dilakukan berdasarkan dua parameter yaitu pengujian berdasarkan jumlah data *item* dan pengujian berdasarkan tingkat *sparsity*. Pada setiap parameter dilakukan percobaan sebanyak 5 kali. Berikut merupakan prosedur pengujian untuk tiap-tiap parameter yang digunakan :

1. Berdasarkan Jumlah *Item*.

Proses pengujian ini dilakukan untuk setiap data yang telah ditentukan jumlahnya, yaitu berjumlah 200 *item*, 300 *item*, 550 *item*, 1000 *item*, 1500 *item*, 2000 *item*, 4000 *item*, 8000 *item*.

2. Berdasarkan Tingkat *Sparsity*.

Tahapan pengujian dilakukan dengan mengosongkan *dataset* secara acak dengan tingkat *sparsity* sebanyak 30% 50% dan 60%.

#### 5.2.3.1. Hasil Pengujian Waktu Eksekusi

Setelah dilakukan pengujian dengan beberapa parameter, maka hasil pengujian untuk setiap parameter diperoleh durasi waktu sebagai berikut:

1. Berdasarkan Jumlah *Item*.

Tabel 5.4. Waktu Eksekusi Berdasarkan Jumlah *Item*

Jumlah <i>Item</i>	Percobaan ke-						Rata- rata
	I	II	III	IV	V	VI	
200	2.4576	2.1191	0.9653	2.5655	0.8931	0.6994	1.4269
300	4.0312	2.9346	0.8813	3.6918	1.2582	1.0449	2.307
550	7.7394	5.2092	2.0398	7.0236	2.2632	1.4194	4.2824
1000	8.5758	5.6950	2.2044	7.7611	2.4327	1.4960	4.6941
1500	16.148	10.152	3.9289	15.015	4.3032	2.6020	8.6915
2000	26.124	16.155	6.1143	24.122	6.6153	3.9382	13.844
4000	60	37.326	14.077	57.236	15.171	13.831	32.940
8000	60	60	29.732	60	53.531	38.928	50.365

Tabel 5.4 menggambarkan waktu yang dibutuhkan untuk setiap pengujian yang dilakukan. Dari hasil pengujian secara keseluruhan, *trend* waktu eksekusi yang dibutuhkan meningkat sejalan dengan banyaknya *item* yang diolah. Sehingga diperoleh waktu tercepat yaitu ketika memproses *dataset* yang berjumlah 200 *item* yaitu dengan rata-rata 1.4269 detik dan waktu terlama yaitu ketika memproses *dataset* yang berjumlah 8000 *item* yaitu dengan rata-rata 50.365 detik.

## 2. Berdasarkan tingkat *sparsity*.

Tabel 5.5. Waktu Eksekusi Berdasarkan Tingkat *Sparsity*

Tingkat <i>Sparsity</i>	Percobaan ke-						Rata- rata
	I	II	III	IV	V	VI	
30%	2.4849	2.1203	0.9712	2.5345	0.8927	0.6860	1.6149
50%	2.1437	1.9410	0.9125	0.7983	0.7983	0.6030	1.5625
60%	0.9268	0.8302	0.2755	0.9818	0.2494	0.3517	0.6732

Tabel 5.5 menunjukan pengaruh tingkat *sparse item* pada waktu yang dibutuhkan untuk mengeksekusi sistem. Pada pengujian ini diketahui bahwa tingkat *sparsity dataset* berbanding terbalik dengan waktu eksekusi yang dibutuhkan. Dimana semakin banyak *item* yang belum di-*rating* (*sparsity* tinggi) maka waktu yang dibutuhkan untuk mengeksekusi semakin kecil. Hal tersebut terlihat pada pengujian ketika tingkat *sparsity*-nya mencapai 60% waktu yang dibutuhkan untuk mengeksekusi sistem rata-rata 0.6732 detik.

### 5.2.4. Analisa Hasil Pengujian

Dari hasil pengujian yang telah dilakukan dengan metode *black box* dapat diperoleh kesimpulan bahwa sisi fungsionalitas dari sistem yang dibangun telah memberikan hasil yang diharapkan, bahwa sistem telah sesuai dengan analisa dan perancangan.

Pengujian tingkat akurasi pada sistem rekomendasi buku perpustakaan dengan metode *collaborative filtering* yang telah dibangun mampu mengatasi kekosongan data dengan tingkat *sparsity* sebanyak 60%, dimana nilai MAE yang

dihasilkan rata-rata sebesar 0.884307. Sedangkan hasil terbaik diperoleh ketika tingkat *sparsity* sebanyak 30%, dimana nilai MAE yang dihasilkan rata-rata sebesar 0.715677.

Merujuk pada hasil penelitian lain (Sarwar,2001) dengan metode yang sama menunjukkan nilai MAE yang dihasilkan berkisar 0.73 - 0.85. Penelitian lain yang dilakukan (Budianto,2012) menunjukkan nilai MAE 0,8361 ketika *sparsity* mencapai 70%, Budianto menggunakan teknik *smoothing* untuk mengatasi masalah *sparsity*, pada teknik ini sistem diizinkan untuk mengisi nilai *rating* yang masih kosong. Hal inilah yang menyebabkan pada penelitian tersebut sistem mampu menampung *sparsity* hingga 70%. Penelitian yang dilakukan penulis tidak menggunakan *smoothing* dan dapat mengatasi *sparsity* hingga 60%. Dengan demikian tingkat akurasi dari penelitian ini tidak terlalu buruk meskipun menggunakan *dataset* yang berbeda.

Dari pengujian waktu eksekusi diketahui bahwa jumlah *item* mempengaruhi waktu eksekusi. Peningkatan jumlah *item* yang diiringi dengan peningkatan jumlah *item* yang di-*rating* akan menghasilkan perbedaan waktu yang signifikan. Terlihat pada *dataset* yang berjumlah 200 *item* rata-rata waktu eksekusi sistem rata-rata 1.4269 detik dan ketika jumlah *item* ditambah hingga berjumlah ribuan waktu eksekusi meningkat seperti pada data yang berjumlah 1000 *item* waktu eksekusi yang diperlukan rata-rata 4.6941 detik. Waktu pemrosesan yang paling lama yaitu ketika sistem mengolah data sebanyak 8000 *item* dimana waktu eksekusi yang dibutuhkan rata-rata 50.365 detik.

Selain jumlah *item*, tingkat *sparsity* juga mempengaruhi waktu eksekusi sistem. Semakin tinggi tingkat *sparsity* maka waktu eksekusi yang dibutuhkan semakin kecil dan sebaliknya semakin rendah tingkat *sparsity* maka waktu eksekusi yang dibutuhkan akan semakin meningkat. *Dataset* dengan tingkat *sparsity* rendah memang membutuhkan waktu eksekusi yang cukup lama dibandingkan dengan tingkat *sparsity* yang lain, namun akan menghasilkan tingkat akurasi yang tinggi karena banyaknya data *rating* yang diproses. Perbandingan antara nilai MAE dan waktu eksekusi dapat dilihat pada Tabel 5.6.

Tabel 5.6. Perbandingan Rata-Rata MAE dan Waktu Eksekusi

Tingkat <i>Sparsity</i>	MAE	Waktu (s)
30%	0.715677	1.6149
50%	0.753503	1.5625
60%	0.84401	0.6732

#### 5.2.5. Kesimpulan Pengujian

Dari pengujian-pengujian yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut :

1. Berdasarkan jawaban kuisioner yang diberi mahasiswa sistem rekomendasi buku perpustakaan menghasilkan rekomendasi buku yang relevan sebanyak 30%.
2. Dari segi keakuratan hasil rekomendasi, nilai MAE dipengaruhi oleh tingkat *sparsity* data. Semakin tinggi *sparsity* data maka nilai MAE akan semakin tinggi. Pada sistem ini dapat mengatasi *sparsity* data hingga 60% dengan nilai MAE 0.884307.
3. Dari segi waktu eksekusi, banyaknya waktu yang dibutuhkan untuk menghasilkan rekomendasi dipengaruhi oleh jumlah data yang diproses serta tingkat *sparsity* data. Semakin tinggi tingkat *sparsity* maka waktu eksekusi yang dibutuhkan semakin kecil dan sebaliknya semakin rendah tingkat *sparsity* maka waktu eksekusi yang dibutuhkan akan semakin meningkat.

## BAB VI

### PENUTUP

#### 6.1. Kesimpulan

Berdasarkan analisis, perancangan, implementasi dan pengujian sistem rekomendasi buku perpustakaan dengan menggunakan metode *collaborative filtering*, maka dapat diambil beberapa kesimpulan diantaranya adalah sebagai berikut:

1. Secara umum, rancang bangun sistem rekomendasi ini berhasil memberikan rekomendasi buku kepada mahasiswa dan berdasarkan UAT 30% dari buku yang direkomendasikan relevan dengan kebutuhan *user*.
2. Hasil pengujian yang ditunjukkan pada Tabel 5.3 membuktikan bahwa sistem dapat mengatasi tingkat *sparsity* data hingga 60% dengan nilai MAE 0.884307.
3. Waktu eksekusi yang dibutuhkan dipengaruhi oleh jumlah data dan tingkat *sparsity*. Pada Tabel 5.4 dapat disimpulkan semakin besar jumlah data yang digunakan maka semakin besar juga waktu eksekusi yang dibutuhkan sedangkan untuk tingkat *sparsity* pada Tabel 5.5 dapat disimpulkan semakin tinggi *sparsity* maka waktu eksekusi yang dibutuhkan semakin kecil.
4. Kekurangan pada sistem ini yaitu ketika *dataset* memiliki *sparsity* yang melebihi 60%, sistem akan menghasilkan nilai MAE yang relatif lebih besar.
5. Pencarian pada sistem ini hanya menggunakan *query sql*. Karena menggunakan *query sql* maka hasil pencarian yang diberikan hanya berdasarkan judul dan daftar isi yang sesuai dengan kata kunci.

## 6.2. Saran

Adapun saran-saran yang diajukan oleh penulis untuk perbaikan rancang bangun aplikasi steganografi ini adalah:

1. Memberi penilaian yang berbeda untuk peminjaman atau pencarian yang dilakukan pada waktu lampau dengan peminjaman atau pencarian yang dilakukan pada saat ini. Hal tersebut dilakukan agar buku-buku yang direkomendasikan adalah buku-buku yang benar-benar relevan dan sesuai dengan minat dan kebutuhan *user* pada saat ini.
2. Berdasarkan kesimpulan yang dijelaskan di atas bahwa rancang bangun sistem ini masih kurang tahan terhadap *sparsity* data yang melebihi 60%. Oleh sebab itu, penulis menyarankan untuk penelitian selanjutnya dapat menggunakan teknik *smoothing* pada *dataset* sehingga penerapan teknik *smoothing* pada keadaan ini juga menyebabkan data yang digunakan untuk melakukan proses prediksi lebih banyak diisi oleh mesin rekomendasi dari pada jumlah *rating* sebenarnya yang di-*rating* langsung oleh *user*.
3. Pada proses pencarian buku sebaiknya menggunakan metode-metode yang berhubungan dengan sistem temu kembali informasi agar nilai *rating* yang diperoleh melalui pencarian lebih optimal karena hasil pencarian yang lebih berkualitas.

## DAFTAR PUSTAKA

- Ampazis, Nicholas. *Collaborative Filtering via Concept Decomposition on the Netflix Dataset*. [Online] Available  
[http://www.feeds2.com/netflix/Ampazis\\_ECAI08.pdf](http://www.feeds2.com/netflix/Ampazis_ECAI08.pdf), diakses 13 Maret 2012.
- Babbie, Earl R. *The Practice of Social Research*. 2012. [Online] Available  
<http://books.google.co.id/books?id=k-aza3qSULoC&dq=Likert+and+Guttman+scaling>, diakses 6 November 2012.
- Budianto, Teguh. *Rancang Bangun Movie Recommender System Dengan Metode Cluster-Based Smoothing Collaborative filtering*. Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau, 2012.
- Claypool, Mark, Phong Le, Makoto Waseda, David Brown. *Implicit Interest Indicators*. Computer Science Department, Worcester Polytechnic Institute, 2001.
- Hakim, Irfan Aris Nur. *Sistem Rekomendasi Film Berbasis Web Menggunakan Metode Item-Based Collaborative Filtering Berbasis K-Nearest Neighbor*. Laporan skripsi sarjana, Program Studi Ilmu Komputer, Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam Universitas Pendidikan Indonesia, 2010.
- Manning, Christopher D., Prabhakar Raghavan, Hinrich Schütze An. *Introduction to Information Retrieval*. Cambridge University Press. 2008.
- Pazzani ,Michael J., Daniel Billsus. *Content-based Recommendation System*. Rutgers University, 2007.
- Purwanto, Ari. *Metode Analisis Rekomendasi Pada Sistem Rekomendasi (Contoh Kasus Pemanfaatan Pada Biro Wisata)*. Program Studi Ilmu Komputer, Fakultas Pendidikan Matematika Dan Ilmu Pengetahuan Alam Universitas Pendidikan Indonesia, 2009.

- Sanjung, Ariyani. *Perbandingan Semantic Classification dan Cluster-based Smoothed pada Recommender System berbasis Collaborative filtering*. Fakultas Teknik Informatika, Institut Teknologi Telkom, 2011.
- Sarwar, Badrul, George Karypis, Joseph Konstan, Jhon Riedl. *Item-based Collaborative Filtering Recommendation Algorithms*. Departement of Computer Science and Engineering, University of Minnesota. 2001.
- Soedjianto, Felicia, Tanti Oktavia, James Arthur Anggawinata. *Perancangan dan Pembuatan Sistem Perencanaan Produksi (Studi Kasus Pada pt. Vonita Garment)*. Seminar Nasional Aplikasi Teknologi Informasi (SNATI).2006.
- ‘Uyun\*, Shofwatul, Imam Fahrurrozi, Agus Mulyanto. *Item Collaborative Filtering untuk Rekomendasi Pembelian Buku secara Online*. Program Studi Teknik Informatika, Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Sunan Kalijaga, 2011.
- Wibowo, Adi, Andreas Handojo, Minardi Taliwang. *Recommender System di Perpustakaan Universitas Kristen Petra menggunakan Rocchio Relevance Feedback dan Cosine Similarity*. Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra, 2010.
- Wimmer, D. Roger, Joseph R. Dominick. *Mass Media Research: An Introduction*. Halaman 55. 2006. [Online] Available <http://books.google.com/books?id=ay5lKmAw2UcC>, diakses 4 November 2012.
- Wiranto, Edi Winarko. *Konsep Multicriteria Collaborative Filtering Untuk Perbaikan Rekomendasi*. Seminar Nasional Aplikasi Teknologi Informasi (SNATI).2010.
- . “Pengembangan Digital Library Dalam Mendukung World Class University ” [Online] Available <http://pustaka.uns.ac.id/?menu=news&option=detail&nid=243> diakses 28 Februari 2012.